

UNIVERSIDAD POLITÉCNICA DE CATALUÑA

DEPARTAMENTO DE FÍSICA

GRUPO DE ASTRONOMÍA Y ASTROFÍSICA



ISFAA: IMPLICIT SPH
FOR ASTROPHYSICAL APPLICATIONS



UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Memoria presentada por
Jose Antonio Escartín Vigo
para optar al grado de
Doctor en Ciencias

Barcelona, Diciembre de 2015

PROGRAMA DE FÍSICA COMPUTACIONAL Y APLICADA

Memoria presentada por **Jose Antonio Escartín Vigo**
para optar al grado de Doctor en Ciencias

DIRECTORES DE LA TESIS

Jose Antonio Escartín Vigo

Dr. Domingo García Senz

Dr. Eduardo Bravo Guil

*A mis grandes apoyos,
Antonio, Pilar y Helena.*

Todos somos muy ignorantes, lo que ocurre es que no todos ignoramos las mismas cosas.

ALBERT EINSTEIN

Es curioso, cuanto más practico más suerte tengo.

ARNOLD PALMER

Aquellos que no recuerdan el pasado, están condenados a repetirlo.

JORGE SANTAYANA

Agradecimientos

Si hecho la vista atrás, el documento que aquí presento ha sido el reto más difícil de mi vida. Durante los últimos años, para concluir este arduo proyecto he hecho muchos sacrificios y sobre todo he tenido el apoyo de mucha gente de mi entorno. Como dice el refrán “de bien nacido es ser agradecido” y con el ánimo de no olvidarme de nadie, quiero agradecer:

- A Antonio Escartín, Pilar Vigo y Helena Sanz, por todo su apoyo en los buenos momentos y por sus ánimos en los malos.
- A mis amigos Guillermo Altabella, Guillermo Santamaría, Luis Delgado y Pablo Panzano que siempre han estado allí cuando he necesitado despejarme de este absorbente trabajo. Y mis compañeros en la UPC que con el tiempo puedo decir que también han llegado a ser amigos, Rubén Cabezón, Oscar Lorente, Pedro Talavera y Antonio Relano.
- A mis directores de tesis, Domingo García-Senz y Eduardo Bravo, por la cantidad de tiempo que me han dedicado y todo el esfuerzo, trabajo y empeño que han puesto para poder concluir la tesis que aquí presento.
- A Jordi José por ceder la financiación necesaria para comprar la máquina de computación, a través del proyecto “ESF EUROCORES Program Eurogenesis (MICINN EUI2009-04167)”, que ha sido imprescindible para probar el código.
- A Enrique García-Berro por coordinar los proyectos “AYA2008-04211” y “AYA2011-23102” que me han servido de financiación para asistir a los congresos necesarios para publicitar mi trabajo.
- Y por último al AGAUR de la “Generalitat de Catalunya” por la financiación que he recibido a través del programa de “Formació d’investigadors” FI.

“Con el soporte del *Comisionado para las Universidades e Investigación del Departament d’Innovació, Universitats i Empresa* de la *Generalitat de Catalunya* y el *Fondo Social Europeo*.”

Índice general

0. Prefacio	1
I Memoria del proyecto	
1. Introducción	7
2. Smoothed Particle Hydrodynamics (SPH)	11
2.1. Fundamentos	13
2.1.1. Kernel	15
2.1.2. Longitud de suavizado	18
2.1.3. Integración y evolución temporal	20
2.2. Ecuaciones	21
2.2.1. Aproximación de las derivadas en las ecuaciones diferenciales	22
2.2.2. Hidrodinámica de Euler: SPH estándar	23
2.2.3. Formalismo Euler-Lagrange: SPH variacional	27
2.2.4. Auto-gravedad	31
2.2.5. Viscosidad artificial	39
2.2.6. Conducción térmica	41

2.2.7. Ecuación de estado (EOS)	42
2.2.8. Contornos	44
2.2.9. Resumen de las ecuaciones que se utilizarán en ISFAA	46
2.3. Esquemas	47
2.3.1. Tipos de métodos: Directos e iterativos	47
2.3.2. Tipos de esquemas: Explícitos e implícitos	49
2.3.3. SPH explícito: Integración centrada mediante <i>predictor-corrector</i>	50
2.3.4. SPH implícito: Integración <i>Newton-Raphson</i>	54
3. Implicit SPH For Astrophysical Applications (ISFAA)	73
3.1. Esquema de trabajo: Flujo de ejecución	75
3.1.1. Parametrización	76
3.2. Iteración	77
3.2.1. Refinando el esquema Newton-Raphson	78
3.2.2. Actualización del modelo	78
3.3. Refinamiento	81
3.3.1. Calcular ecuaciones y derivadas	82
3.3.2. Calcular las variaciones de las variables	84
3.3.3. Comprobar la convergencia	90
3.4. Coste computacional	91
3.5. Funcionalidades adicionales	92
4. Tests básicos	93
4.1. Explosión puntual	94

4.2. The wall heating shock	102
4.3. Inestabilidad de Rayleigh-Taylor	106
4.4. Caída libre	113
4.5. Evolución de una onda térmica	122
5. Tests con gravedad	125
5.1. Evolución casi-estática de un sistema próximo al equilibrio	126
5.2. Equilibrio hidrostático de una estrella de tipo solar	132
5.3. Equilibrio hidrostático de una enana blanca masiva ($m \simeq 1.15M_{\odot}$)	137
5.4. Aproximación implícita cuasi-hidrostática	141
6. Conclusiones	147
6.1. Objetivos conseguidos	148
6.2. Avances en el método implícito	149
6.3. Viabilidad del método implícito	149
6.4. Trabajo futuro	150
II Apéndices	
A. Kernel	155
A.1. Kernel Cubic spline: $W(v, h)$	158
A.2. Kernel Sinc: $W^S_n(v, h, n)$	159
B. Derivadas parciales de las ecuaciones físicas	161
C. Física adicional	167

C.1. Viscosidad Artificial (Π_{ij})	167
C.2. Gravedad (g_i)	171
C.3. Conducción térmica (κ)	175
C.4. Damping	177
 D. Ecuación de estado (EOS)	 179
 E. Análisis y convergencia en las ecuaciones	 185
 F. ISFAA y PANGAEA	 193
 G. Aceleradoras gráficas y coprocesadores	 199
 H. Publicaciones	 207
 Bibliografía	 209

Índice de cuadros

2.1. Función $f(x)$: Aproximación de la primera derivada	22
2.2. Función $f(x)$: Aproximación de la segunda derivada	23
2.3. Ecuaciones hidrodinámicas de Euler	24
2.4. Ecuaciones hidrodinámicas de Euler en SPH	27
2.5. Cálculo de la conservación de la energía	31
2.6. Ecuaciones del formalismo Euler-Lagrange en SPH	32
2.7. Gravedad: Ley de Gauss	34
2.8. Discretización de las ecuaciones del SPH	46
2.9. Esquema implícito: Ejemplo con dos ecuaciones y dos incógnitas	58
2.10. SPH implícito: Integración <i>Newton-Raphson</i>	60
2.11. SPH implícito: Esquema completo	61
2.12. SPH implícito: Esquema reducido en velocidad y densidad	69
2.13. Discretización de las ecuaciones del SPH: Esquema cuasí-hidrostático	70
2.14. SPH implícito: Esquema cuasí-hidrostáticoreducido en densidad	71
3.1. Δt : Restricciones generales	80
3.2. Δt : Restricciones en la simulación	80

3.3. Δt : Restricciones en la variación de variables	81
3.4. Variables independientes obligatorias: Posiciones y energía/temperatura	88
3.5. Variables independientes optativas: Velocidades y densidad	89
4.1. Test de Sedov: Tiempo de computación	96
4.2. Test de Noh: Tiempo de computación	104
4.3. Test de Rayleigh-Taylor: Condiciones iniciales	108
4.4. Test de Rayleigh-Taylor: Tiempo de computación	109
4.5. Test de caída libre: Tiempo de computación	116
5.1. Test sistema próximo al equilibrio: Tiempo de computación	129
5.2. Test estrella de tipo solar: Tiempo de computación	132
5.3. Discretización de las ecuaciones del SPH: Esquema cuasí-hidrostático	141
A.1. Kernel Sinc: Derivadas del parámetro v_0	155
A.2. Kernel Sinc: Derivadas de la distancia \mathbf{r}_{ij}	156
A.3. Kernel Cubic Spline: Parámetro Ω	157
A.4. Kernel Cubic Spline: Signos dependiendo del orden de índices i, j	157
A.5. Kernel $W(v, h)$: Derivadas	158
A.6. Kernel Cubic Spline: Derivadas de C	158
A.7. Kernel $\mathbf{W}'(v, h)$: Derivadas	159
A.8. Kernel: Constante de normalización B_n	159
A.9. Kernel Sinc: Derivadas de <i>sinc</i>	160
A.10. Kernel Sinc: Derivadas de S_n)	160

B.1. Ecuación G_{a_k} : Derivadas	163
B.2. Ecuación $G_{\dot{E}}$: Derivadas	164
B.3. Variable e_i : Derivadas	164
B.4. Ecuación G_h : Derivadas	165
B.5. Ecuaciones G_{v_x} , G_{v_y} y G_{v_z} : Derivadas	165
B.6. Ecuación G_{ρ} : Derivadas	166
C.1. Viscosidad artificial: Formulación estándar	168
C.2. Viscosidad artificial: μ_{ij}	169
C.3. Viscosidad artificial: Pi_{ij}	169
C.4. Viscosidad artificial: Ecuación del momento	170
C.5. Viscosidad artificial: Ecuación de la energía	170
C.6. Gravedad según un oscilador armónico	171
C.7. Gravedad directa: Método explícito	172
C.8. Gravedad según la ley de Gauss: 2D (x,y)	173
C.9. Gravedad según la ley de Gauss: 3D (x,y,z)	173
C.10.Gravedad según el potencial gravitatorio: Calculo del potencial gravitatorio	174
C.11.Gravedad según el potencial gravitatorio	175
C.12.Conducción térmica: Ecuación y derivadas	176
C.13.Conducción térmica: Derivadas intermedias I	176
C.14.Conducción térmica: Derivadas intermedias II	177
C.15.Damping	177
D.1. EOS ideal: Conversión entre energía y temperatura	180

D.2. EOS ideal: Derivadas	180
D.3. EOS realista: Derivadas	182
D.4. EOS realista: Factores de relación entre energía y temperatura	184

Índice de figuras

2.1. Longitud de suavizado	20
2.2. Dominio espacial: Distribución de partículas en un árbol cuadrático en 2D	36
2.3. Momento cuadrupolar: Aplicación	37
2.4. Tipos de multipolo	38
2.5. Función de contorno $f(r)$ vs kernel	45
2.6. Dominio espacial y temporal: Discretización en 1D	51
2.7. Discretización semi-implícita: Factor de amplificación	56
2.8. Discretización semi-implícita: Estabilidad numérica	56
2.9. Dominio espacial: Distribución de partículas en 3D según el algoritmo z-order . .	65
3.1. ISFAA: Diagrama de flujo de la secuencia de ejecución	75
3.2. ISFAA: Diagrama de flujo de las iteraciones	77
3.3. ISFAA: Diagrama de flujo de los refinamientos	82
3.4. Distribución espacial del árbol octal en 3D	83
3.5. Métodos para resolver sistemas de ecuaciones lineales	86
3.6. Matriz dispersa: Ejemplo de compresión CSR	87

4.1. Test de Sedov: Tiempos de computación de la simulación	97
4.2. Test de Sedov: Tiempos de computación de la simulación	98
4.3. Test de Sedov: Perfiles de densidad, velocidad y presión	99
4.4. Test de Sedov: Evolución temporal	100
4.5. Test de Sedov: Conservación de la energía y momento lineal	101
4.6. Test de Noh: Tiempos de computación de la simulación	103
4.7. Test de Noh: Perfiles de densidad, velocidad y presión	104
4.8. Test de Noh: Evolución temporal	105
4.9. Test de Noh: Conservación de la energía y momento lineal	106
4.10. Test de Rayleigh-Taylor: Perfiles a $t = 0$ s	107
4.11. Test de Rayleigh-Taylor: Tiempos de computación de la simulación	110
4.12. Test de Rayleigh-Taylor: Perfiles de densidad, velocidad y presión	111
4.13. Test de Rayleigh-Taylor: Evolución temporal	112
4.14. Test de Rayleigh-Taylor: Conservación de la energía y momento lineal	113
4.15. Test de caída libre: Tiempos de computación de la simulación	115
4.16. Test de caída libre: Perfiles de densidad, velocidad y presión	117
4.17. Test de caída libre: Evolución temporal 2D	118
4.18. Test de caída libre: Evolución temporal 2D	119
4.19. Test de caída libre: Conservación de la energía y momento lineal	120
4.20. Test de caída libre: Conservación de la energía y momento lineal	121
4.21. Test onda térmica: Posiciones de las partículas	123
4.22. Test onda térmica: Evolución temporal de la T	124

5.1. Test sistema próximo al equilibrio: Tiempos de computación de la simulación . . .	128
5.2. Test sistema próximo al equilibrio: Perfiles	129
5.3. Test sistema próximo al equilibrio: Evolución temporal	130
5.4. Test sistema próximo al equilibrio: Conservación de la energía	131
5.5. Test sistema próximo al equilibrio: Conservación momento lineal y angular . . .	131
5.6. Test estrella de tipo solar: Tiempos de computación de la simulación	133
5.7. Test estrella de tipo solar: Perfiles	134
5.8. Test estrella de tipo solar: Evolución temporal	135
5.9. Test estrella de tipo solar: Conservación de la energía	136
5.10. Test estrella de tipo solar: Conservación momento lineal y angular	136
5.11. Test estrella enana blanca: Evolución temporal	138
5.12. Test estrella enana blanca: Gradiente de presión Vs gravedad	139
5.13. Test estrella enana blanca: Energía	140
5.14. Test estrella enana blanca: Gradiente de presión Vs gravedad	140
5.15. Modelo cuasí-hidrostático: Tiempos de computación de la simulación	143
5.16. Modelo cuasí-hidrostático: Evolución temporal	144
5.17. Modelo cuasí-hidrostático: Conservación de la energía	145
5.18. Modelo cuasí-hidrostático: Perfiles	146
E.1. Criterio de Courant	187
F.1. PANGEA: Máquina de computación	198
F.2. PANGEA: Conexiones a la red	198
G.1. Comparativa entre CPUs y GPUs: Aplicaciones	200

G.2. Comparativa entre CPUs y GPUs: <i>Speedup</i> y <i>performance</i>	201
G.3. Comparativa entre CPUs y GPUs: Proyección	202
G.4. Intel Xeon Phi: Knight Corner - Múltiples tarjetas	203
G.5. Intel Xeon Phi: Knight Corner - Diagrama	203
G.6. Intel Xeon Phi: Knight Landing - Diagrama	204
G.7. Lista <i>top500</i> 45th(2015): List	204
G.8. Lista <i>top500</i> 45th(2015): Distribución de aceleradoras	205
G.9. Lista <i>top500</i> 44th(2014): Núcleos por socket y distribución de aceleradoras	206

Capítulo 0

Prefacio

La simulación mediante ordenador es una de las herramientas básicas de la Astrofísica moderna. Su uso permite estudiar sistemas complejos, como las estrellas, las galaxias, o fenómenos asociados, como explosiones de supernova, con un grado de realismo suficiente para comparar los resultados de dichas simulaciones con datos observacionales cada vez de mayor calidad. Esta contrastación entre resultados de simulaciones y datos permite a su vez refinar las simulaciones y, de esta manera, avanzar en la comprensión del Cosmos y la predicción del resultado de nuevos experimentos y/o observaciones astronómicas.

En las simulaciones se deben incorporar todos los mecanismos y procesos físicos que influyan significativamente en los resultados que se quieran comparar con los datos. Una dificultad que surge con frecuencia es que procesos astrofísicos imprescindibles en la simulación involucran escalas de tiempo que difieren en varios órdenes de magnitud.

Los procesos de gran escala temporal son imposibles de tratar con enfoques explícitos ya que estos se encuentran limitados, en su paso de tiempo máximo, por la restricción conocida como condición de *Courant-Friedrichs-Lewy*, 1928 [22]. Sería necesario un número de iteraciones explícitas prohibitivas para reproducir este tipo de sistemas e incluso si fuese posible la acumulación de errores de redondeo pervertiría la solución. Para trabajar con las escalas temporales más grandes se deben utilizar enfoques implícitos. Se puede citar como ejemplo *Henye y colaboradores*, 1959 [37], donde se introducen diferencias finitas implícitas en los cálculos de evolución estelar cuasi-estáticos (reacciones nucleares e hidrodinámica), o el trabajo más reciente basado en el método de los volúmenes finitos *Viallet y colaboradores*, 2011 [100] propuesto para resolver los procesos de evolución estelar.

Para utilizar los enfoques implícitos se genera un sistema de ecuaciones algebraicas acopladas, habitualmente resuelto con un esquema de Newton-Raphson y compuesto por todas las

ecuaciones de cada uno de los puntos de resolución del modelo. El coste computacional de resolución aumenta sustancialmente con el número de incógnitas que han de determinarse a cada paso de tiempo. Por ejemplo, dado un mismo problema, si lo tratamos con un enfoque implícito y su resolución es del orden de 10^4 (decenas de miles de partículas), tratado con un enfoque explícito su resolución podría ser del orden de 10^6 (varios millones de partículas). Debido a esta restricción de tamaño, la hidrodinámica implícita históricamente ha sido aplicada a sistemas en una sola dimensión, por ejemplo en los códigos hidrodinámicos con simetría esférica para simulación de supernovas que se pueden encontrar en *Schinder y Bludman, 1989* [87], *Swesty, 1995* [98], *Yamada, 1997* [103] y *Liebedörfer y colaboradores, 2002* [48].

Últimamente se han generado versiones implícitas para enfoques eulerianos o de mallas, en el artículo de *Kifonidis y Müller* [46] se describen los avances con estos métodos. En este artículo se nombran casos especialmente difíciles de tratar con los métodos explícitos como la propagación de ondas de choque que se encuentran en los flujos de: 1) x-ray bursts (*Lin y colaboradores, 2006* [49]), 2) la fases previas que conducen a la ignición en las novas clásicas (*Kercek y colaboradores, 1999* [45] y *Glasner y colaboradores, 2007* [28]) y 3) las fases previas a la ignición de las supernovas de tipo Ia (*Höflich y Stein, 2002* [40], *Zingale y colaboradores, 2009* [106]). Estos flujos son inicialmente subsónicos pero debido a la liberación de energía de los procesos termonucleares evolucionan convirtiéndose en flujos supersónicos. En estos casos, pueden coexistir en el mismo dominio computacional regiones subsónicas (con flujos casi incompresibles) con regiones supersónicas (con flujos muy compresibles). Los métodos explícitos, utilizados en flujos compresibles, en estas situaciones tienen un uso muy limitado. Por ejemplo para estas situaciones, y solamente para sistemas con números de *Mach* bajos, se puede recurrir al método euleriano desarrollado por *Almgren y colaboradores, 2006-2008* [2] [3] [1], denominado MAESTRO, que puede manejar simultáneamente regímenes con flujos compresibles e incompresibles.

Para la implementación también se puede utilizar enfoques lagrangianos como el suavizado de partículas hidrodinámicas denominado “Smooth Particle Hydrodynamics” ó *SPH* (*Monaghan, 2005* [61]). Esta técnica se viene aplicando con éxito al campo de la astrofísica, la cosmología y diferentes problemas de la física de fluidos. El SPH integra las ecuaciones de la dinámica de fluidos en cada punto del formalismo lagrangiano (denominado partícula por tener una masa asociada) calculando velocidad, posición, densidad y presión como una interpolación de los valores de las partículas vecinas. Los métodos lagrangianos, a diferencia de los eulerianos, no necesitan de una malla regular que cubra la totalidad del espacio de integración, por tanto, la memoria y el tiempo de cálculo no se desperdician en la resolución de espacios vacíos. Los fluidos se descomponen en un conjunto de partículas donde podemos tratar numéricamente de forma mas sencilla el movimiento en tres dimensiones derivado de las fuerzas de presión y auto-gravedad. Es especialmente idóneo para tratar problemas no simétricos, que incluyen espacios vacíos ya que estos no influyen en el cálculo.

Es necesario tener en cuenta también las desventajas del método como:

- El desorden de las partículas afecta a la precisión del cálculo.
- En los choques es necesario introducir viscosidad artificial.
- En algunos problemas las condiciones de contorno pueden ser complicadas de implementar.
- En la interpolación se utilizan funciones con simetría esférica y la distribución de vecinos debería de ser isotrópica, si esto no se cumple las partículas pueden alinearse en discos o láminas.

La técnica SPH ha sido históricamente utilizada para simular multitud de fenómenos en diferentes campos de la ciencia, como en:

- Astrofísica: Simulación de supernovas (*Bravo y García-Senz, 2008* [13]) , interacción y colisiones de estrellas (*Aznar-Siguán y colaboradores* [6], *Lorén-Aguilar y colaboradores* [53], *Guerrero y colaboradores, 2004* [31]) , relatividad especial (*Rosswog, 2010* [79]) , magnetohidrodinámica (*Camenzind, 2005* [18], *Price y Bate, 2007* [72]) , ...
- Cosmología: Interacción galáctica (*Berczik, 2000* [10]) , simulación cosmológica (*Springel* [93]) , enriquecimiento químico (*Wiersma y colaboradores, 2009* [102]) , ...
- Dinámica de gases y fluidos: Movimiento, interacciones y turbulencias (*Gómez-Gesteira* [29]).
- Ciencia de los Materiales: Impactos, colisiones y fragmentaciones (*Johnson y colaboradores* [43]).

Resumiendo, de todo lo dicho anteriormente se deduce que el método SPH es ideal para conseguir los siguientes objetivos pretendidos en esta tesis:

1. Partiendo de la base del trabajo previo de *Knapp, 2000* [47], se construirá un código SPH implícito que simule procesos astrofísicos.
2. Durante el desarrollo se analizará si es viable y competitivo con la tecnología existente utilizar códigos SPH implícitos.
3. Se establecerán las limitaciones y posibilidades del código SPH implícito.
4. Se realizarán tests que comprueben la validez del código y se aplicará a algunos casos prácticos donde el uso del código SPH implícito obtenga ventajas respecto del uso del código SPH explícito.

Parte I

Memoria del proyecto

Capítulo 1

Introducción

Los esquemas implícitos hidrostáticos descritos con simetría esférica fueron uno de los pilares de la computación astrofísica de la segunda parte del siglo pasado. El uso generalizado de los llamados códigos Henyey (*Henyey y colaboradores, 1964* [36]), fueron cruciales para entender la evolución estelar durante los etapas asociadas a la combustión de Hidrógeno y Helio. Por contra los códigos hidrodinámicos explícitos preferidos para reproducir eventos muy dinámicos, fenómenos que se desarrollan en periodos muy cortos de tiempo, como las explosiones de novas o supernovas. Hoy en día el rendimiento de los computadores es lo suficientemente grande como para permitir detalladas simulaciones hidrodinámicas en 3D. En este sentido, un código hidrodinámico muy utilizado es el suavizado de partículas hidrodinámicas (“Smoothed Particle Hydrodynamics” ó SPH), que es un método de naturaleza Lagrangiana. Hasta ahora el SPH se ha aplicado al tratamiento de fenómenos dinámicos y por lo tanto, los esquemas de integración explícitos se prefieren como un marco para implementar dicho método. Sin embargo se debe reconocer la importancia que podría tener un esquema implícito de integración para simular múltiples problemas astrofísicos multidimensionales que requieren el uso de pasos de tiempo largos, comparados con el paso de tiempo de Courant (*Courant y colaboradores, 1967* [22]). Por ejemplo, muy poco se conoce de las aplicaciones astrofísicas relacionadas con la evolución casi-hidrostática, como: 1) las etapas pre-explosivas que ocurren en las novas o 2) las etapas previas que conducen a la explosión de una enana blanca, un fenómeno conocido como explosión de supernovas termonucleares o de tipo Ia (*Hillebrandt y Niemeyer, 2000* [39] y *Höflich y Stein, 2002* [40]). Dichas etapas puede durar varios miles de años haciendo inútil la hidrodinámica explícita, donde el tiempo de Courant se sitúa por debajo del segundo.

La técnica de aproximación numérica SPH fue introducida por *Lucy, 1977* [54] e independientemente por *Gingold y Monaghan, 1977* [62]. Es un método de simulación del movimiento de los fluidos basado en la hidrodinámica Lagrangiana. Consiste en una malla de puntos donde las coordenadas se mueven con el sistema. Cada punto tiene asociadas una serie de propiedades

físicas locales, incluida la masa, que varían con el tiempo, por lo que en adelante lo identificaremos como pseudo-partículas (o partículas SPH). Estas propiedades del fluido se calculan como la suma interpolada de las propiedades de las partículas de su entorno. Permite una implementación en tres dimensiones de forma mucho más sencilla que los métodos Eulerianos y por construcción conserva exactamente la masa, el momento lineal y el momento angular del sistema. El SPH variacional (*Rosswog, 2009* [78]) incluye en las ecuaciones el gradiente de la longitud de suavizado descrito en el capítulo 2 y permite una mejor conservación de la energía y entropía respecto al SPH básico.

Inicialmente fue descrita para escenarios astrofísicos y desde entonces se han mejorado mucho sus características y se ha aplicado a muchos otros campos de la ciencia y la ingeniería. Durante décadas el método SPH ha sido evolucionado por muchos autores (*Benz, 1986* [9]; *Hernquist y Katz, 1989* [38]; *Monaghan y Lattanzio, 1985* [63]; *Monaghan, 1989* [57],...) hasta componer el método que hoy en día conocemos. Estos artículos han tratado ampliamente una gran diversidad de escenarios hidrodinámicos, y cada cierto tiempo aparecen revisiones de los avances en las características del método: *Monaghan, 1992* [58]; *Monaghan, 2005* [61]; *Rosswog, 2009* [78], *Springel, 2010* [95] y *Perego y colaboradores, 2014* [70]. Estos resúmenes agrupan las mejoras del método originadas por la multitud de publicaciones de los grupos de investigación que utilizan la técnica SPH para sus trabajos.

Hasta ahora, debido a la complejidad algebraica inherente a este tipo de métodos y al esfuerzo computacional requerido, pocos autores que trabajen con SPH han investigado sobre una versión implícita (*Knapp, 2000* [47]) para que sea aplicable en escenarios multidimensionales casi-hidrostáticos. En una versión implícita, las propiedades del siguiente paso de tiempo dependen de los valores de las variables desconocidas en dicho paso de tiempo y por tanto todas han de ser calculadas simultáneamente. La consecuencia es que todo el sistema de ecuaciones se ha de resolver conjuntamente realizando la inversión de una matriz dispersa enorme (la matriz tiene un tamaño de $(n \cdot var)^2$, siendo n el número de partículas y var el número de variables independientes del sistema).

En su tesis, *Knapp* propone un método implícito basado en resolvedores iterativos de Krylov y correcciones de Newton-Raphson. Su método fue desarrollado en el año 2000 a partir del código SPH explícito SPHINX de *Los Alamos National Laboratory* y utilizaba una física simplificada, que entre otras cosas solo incluye las ecuaciones básicas del SPH y no contempla gravedad ni viscosidad artificial. Además, en sus tests y aplicaciones, debido a las limitaciones computacionales del momento, incluye muy pocas partículas. Por tanto, su método es insuficiente para describir escenarios astrofísicos complejos.

Por otro lado, la motivación para este proyecto parte de la experiencia que tiene el “Grupo d’Astronomía i Astrofísica” (GAA) de la UPC en el desarrollo de simulaciones mediante SPH

explícitos y que disponer de un código implícito supondría la posibilidad de estudiar escenarios casi-hidrostáticos que cubran las fases previas de las explosiones en novae y supernovas, algo de mucho interés en nuestro grupo.

El objetivo de esta tesis es detallar las principales características de un SPH implícito paralelo al que hemos denominado ISFAA (Implicit SPH for Astrophysical Applications), ampliando el trabajo de Knapp e incluyendo un esquema físico más actual que incorpora la última implementación del SPH, la viscosidad artificial, la gravedad y la conductividad térmica.

Dado el enorme esfuerzo que supone construir un código SPH implícito, se pretende que en el futuro su utilidad se extienda al mayor número posible de escenarios astrofísicos. Con este fin se ha optado por un diseño modular que separe el tratamiento de las ecuaciones evolutivas básicas, de las propiedades del material (ecuación de estado, viscosidad artificial, etc.). Además en el futuro la arquitectura modular permitirá incluir ingredientes físicos todavía no implementados para poder tratar otros problemas astrofísicos casi-hidrostáticos como: las fases avanzadas de la evolución estelar, X-ray bursts, etc.

El documento que aquí se presenta, se encuentra dividido en los siguientes capítulos:

1) Primeramente, en el capítulo 2, se describe la técnica SPH y toda la teoría necesaria para desarrollar una versión implícita.

2) En el capítulo 3, se desarrolla la implementación del código ISFAA, haciendo hincapié en las características específicas de este código, como:

- Utilizar el moderno formalismo de SPH conservativo basado en el principio variacional.
- Usar eficientes bibliotecas paralelas, como PARDISO o WSMP, para la resolución de matrices dispersas derivadas de sistemas de ecuaciones.
- Incorporar la física necesaria para manejar los problemas comunes de astrofísica en tres dimensiones. Como por ejemplo, ecuaciones de estado complejas, gravedad, viscosidad artificial y conductividad térmica.

3) Para comprobar la corrección del código y probar cada uno de los ingredientes físicos, en el capítulo 4, se especifican una serie de test básicos:

- Explosión puntual (test de Sedov).
- The wall heating shock (test de Noh).
- Inestabilidades de Rayleigh-Taylor.

- Caída libre (Free-Fall collapse).
- Expansión de una onda térmica.

que comprueban el buen funcionamiento del código. En estos tests se utilizan un número de partículas pequeño (unas 8 000) comparado con el de los métodos explícitos.

4) Una vez validada la técnica, en el capítulo 5, se pueden encontrar varios test con gravedad que tratan de simular lo siguiente:

- Evolución casi-estática de un sistema próximo al equilibrio (Toy Star en 2D).
- Equilibrio hidrostático de una estrella de tipo solar, construido con un polítropo.
- Equilibrio hidrostático de una enana blanca masiva de $1,15M_{\odot}$.
- Aproximación implícita cuasi-hidrostática. Esto supone la evolución de un sistema casi-estático, en el que las velocidades no se encuentran explícitamente en el modelo y está orientado a demostrar que puede evolucionar en largos intervalos temporales.

5) En el capítulo 6, se comentan las conclusiones finales de la tesis y unas guías de actuación para el trabajo futuro que se prevé realizar para mejorar la versión implícita del SPH.

6) Por último tenemos una serie de apéndices con toda la información complementaria al texto y que resulta necesaria o interesante para entender el funcionamiento del código.

Por tanto el objetivo final es demostrar el potencial del código implícito ISFAA y recalcar que, después del documento de *Knapp 2000* [47], este es el siguiente paso hacia un código SPH implícito generalista.

Capítulo 2

Smoothed Particle Hydrodynamics (SPH)

La aplicación de técnicas numéricas en el campo de la astrofísica ha dado lugar a una rama de investigación completamente nueva centrada en el desarrollo de herramientas y metodologías que han permitido el estudio de problemas complejos pero de forma simulada. El rápido desarrollo de las infraestructuras de supercomputación ha hecho del cálculo hidrodinámico uno de los beneficiarios más evidentes de este esfuerzo.

La implementación numérica de la dinámica de fluidos ha sido habitualmente tratada con mallas, donde las magnitudes físicas se evalúan a través del balance de flujo entre celdas adyacentes y el cálculo de sus derivadas utilizando métodos de diferencias finitas. Estas técnicas han sido profundamente estudiadas y han dado lugar a diferentes formalismos e implementaciones. Sin embargo, muchos escenarios astrofísicos tienen magnitudes físicas que se mueven en un rango muy amplio de ordenes de magnitud, esto exige un comportamiento de alta adaptación de los códigos hidrodinámicos. Este es un reto importante para los métodos basados en malla fija, que se han convertido en uno de los pilares del desarrollo de la nueva generación de códigos de tipo Euleriano o de malla adaptativa (“Adaptative mesh refinement” o AMR, *Berger y Oliger, 1984* [11]; *MacNeice y colaboradores, 2000* [55]) que pueden cambiar dinámicamente la resolución del cálculo tanto espacial como temporalmente, según las necesidades físicas. Los códigos hidrodinámicos AMR tienen implementaciones muy complejas y por tanto se ha de dedicar mucho esfuerzo para desarrollar herramientas y bibliotecas que faciliten la portabilidad y simplifiquen la interacción con el código. Habitualmente los escenarios astrofísicos desarrollan geometrías asimétricas muy deformadas y esto puede ser un problema para los métodos Eulerianos adaptativos (AMR) y no adaptativos, debido a la difusión numérica introducida por el término advectivo.

Esto se puede subsanar mediante una malla que evoluciona con el fluido, para ello debemos dar una descripción Lagrangiana de las ecuaciones de Euler. Esta implementación permite que el término advectivo desaparezca y es una opción muy interesante en geometrías asimétricas. Para ello se deben trabajar con métodos puramente Lagrangianos, donde las mallas son sustituidas por puntos de interpolación con una masa asociada, que siguen al fluido. La técnica de suavizado de partículas hidrodinámicas (“Smooth particle hydrodynamics” o SPH) pertenece a este tipo de códigos. El SPH al no tener malla evita los problemas asociados a la distorsión y la alta resolución necesaria en las zonas de interés.

Las razones principales para elegir este método son las siguientes:

- Por construcción, la conservación del momento lineal, angular, masa y energía es perfecta. Algo que no ocurre en los métodos Eulerianos.
- Puede ser utilizado para simular escenarios astrofísicos muy diversos.
- Existe abundante información detallada sobre la técnica SPH, sus ventajas y sus limitaciones.
- Los años de experiencia que tiene nuestro grupo de investigación con el método SPH. Este ha sido aplicado a supernovas Ia, interacción de estrellas compañeras y choques (como por ejemplo entre enanas blancas o estrellas de neutrones).

La comparación entre los diferentes códigos no es trivial a causa de las diferentes implementaciones: tanto la física, como en los métodos de integración, en el manejo de los choques e incluso en el control de la resolución. Afortunadamente, para muchos escenarios concretos, todos los códigos generan resultados similares. Sin embargo, esto puede cambiar en situaciones críticas y da motivos de estudio de esas diferencias, observando las limitaciones de cada método.

Como se ha detallado anteriormente, el objetivo de la tesis es desarrollar una versión innovadora del código SPH que sirva para aplicar la técnica a escenarios no probados hasta este momento. En las secciones siguientes de este capítulo se hará una descripción genérica del método y se irá desgranando las características del mismo.

Obviamente, el código desarrollado se circunscribe en el contexto tecnológico actual, el cual impone unas limitaciones determinadas en los recursos computacionales.

2.1. Fundamentos

En general los códigos hidrodinámicos se basan en la evaluación de las propiedades físicas del fluido en un conjunto de puntos de la malla, usando diferentes técnicas numéricas (interpolación, diferencias finitas, elementos finitos, ...) para encontrar los valores de las propiedades en cualquier otro punto del sistema.

Los códigos lagrangianos tipo SPH no son mucho más complejos en 3D que en 1D, por contra los códigos eulerianos multidimensionales son muy complejos. En SPH el fluido se discretiza mediante un conjunto de puntos de interpolación con una masa asociada, por ello los llamaremos pseudo-partículas (o partículas SPH). Las propiedades del sistema se evalúan en cada uno de estos puntos como un promedio ponderado de las propiedades físicas de los puntos más cercanos. La contribución de los puntos cercanos se obtiene mediante una función, normalizada a la unidad, que disminuye fuertemente con la distancia. Por tanto, la interpolación es un proceso local que produce un suavizado de las propiedades físicas. Si especificamos la ecuación de estado, la física relevante para el escenario concreto que estamos tratando (gravitación, conducción térmica, ...) y resolvemos las ecuaciones de la hidrodinámica, podemos seguir la evolución del sistema.

La versión suavizada de cualquier magnitud física A es una función de la posición de la partícula y se reconstruye a cada paso de tiempo por medio de una función interpoladora, denominado interpolante de A . Esta función se define como sigue:

$$\langle A(\mathbf{r}) \rangle = \int A(\mathbf{r}') W(|\mathbf{r}' - \mathbf{r}|) d\mathbf{r}' \quad (2.1)$$

donde W es la función de interpolación o kernel, que depende de las coordenadas de la partícula r y de la longitud de suavizado h (veremos este concepto en el apartado 2.1.2).

Esta función kernel es una función real (normalmente con forma gaussiana) que tiende hacia una función delta de Dirac (δ) cuando h tiende a 0 y que tiene soporte compacto, es decir, a partir de una determinada distancia su valor es nulo.

El parámetro h se denomina longitud de suavizado y es una magnitud muy importante del SPH, nos da una idea de la separación media entre partículas y, para cada partícula, establece el número de vecinos que contribuirán a las ecuaciones.

En cálculo numérico, la integral 2.1 puede ser implementada con el sumatorio:

$$A_i(\mathbf{r}_i) = \sum_{j=1}^{N_v} m_j \frac{A_j}{\rho_j} W(\mathbf{r}_i - \mathbf{r}_j, h_i) \quad (2.2)$$

Donde m_j es la masa y ρ_j la densidad de la partícula j . El sumatorio se extiende por los vecinos de la partícula i -ésima. Se dice que una partícula j es vecina de otra i cuando j esta lo

suficientemente cerca de i como para el el valor de $W(\mathbf{r}_i - \mathbf{r}_j, h_i)$ sea distinto de cero (es decir j es vecina de i si se encuentra dentro del rango de interpolación del kernel). El factor m_j/ρ_j es la versión discretizada del volumen diferencial $d\mathbf{r}'$ que aparece en la ecuación 2.1.

Por lo tanto una característica importante de cualquier código SPH es incluir un algoritmo eficiente para encontrar los vecinos de una partícula dada. El método que se usa habitualmente, y que ha sido incluido en el código, fue desarrollado por *Hernquist y Katz, 1989* [38] y se basa en una estructura de árbol jerárquico.

Podemos calcular la derivada de cualquier magnitud según la forma (*Cabezón, 2010* [14]):

$$\nabla A_i = \sum_{j=1}^N \left[\nabla_i \left(\frac{m_j}{\rho_j} A_j \right) W_{ij} + \frac{m_j}{\rho_j} A_j \nabla_i W_{ij} \right] = \sum_{j=1}^N \frac{m_j}{\rho_j} A_j \nabla_i W_{ij} \quad (2.3)$$

donde $\nabla_i W_{ij}$ es el gradiente de $W(\mathbf{r}_{ij}, h)$ con respecto a las coordenadas de la partícula i . El primer término de la suma es cero, por lo tanto, la derivada se calcula directamente como la derivada del kernel, que es una función analítica diferenciable, cuyo valor es conocido. Por tanto podemos evitar muchos de los problemas asociados al cálculo de las derivadas que son comunes a los esquemas de diferencias finitas utilizadas en los métodos de mallas fijas.

En contrapartida, el cálculo de las derivadas a través del kernel puede conducir a soluciones inadecuadas. Por ejemplo, si A es constante debería ser $\nabla A = 0$, no obstante la expresión anterior no da dicho resultado. Para solventar este tipo de problemas se pueden reformular las derivadas numéricas como se explica a continuación.

Observemos las versiones suavizadas de los operadores gradiente, rotacional y divergencia:

$$\nabla A = \frac{1}{\rho} [\nabla(\rho A) - A \nabla \rho] \quad (2.4)$$

$$\nabla \cdot \mathbf{A} = \frac{1}{\rho} [\nabla \cdot (\rho \mathbf{A}) - \mathbf{A} \cdot \nabla \rho] \quad (2.5)$$

$$\nabla \times \mathbf{A} = \frac{1}{\rho} [\nabla \times (\rho \mathbf{A}) + \mathbf{A} \times \nabla \rho] \quad (2.6)$$

Podemos discretizar los términos de la ecuación del gradiente 2.4 de la siguiente forma:

$$\nabla_i(\rho_i A_i) = \sum_j m_j A_j \nabla_i W_{ij} \quad (2.7)$$

$$A_i \nabla_i(\rho_i) = A_i \sum_j m_j \nabla_i W_{ij} \quad (2.8)$$

Si observamos el operador divergencia 2.5 y desarrollamos el primer término:

$$\begin{aligned}
 \nabla_i \cdot (\rho \mathbf{A})_i &= \nabla_i \cdot \left(\sum_j m_j \mathbf{A}_j W_{ij} \right) \\
 &= \sum_j m_j \nabla_i \cdot (\mathbf{A}_j W_{ij}) \\
 &= \sum_j m_j (\mathbf{A}_j \cdot \nabla_i W_{ij} + W_{ij} \nabla_i \cdot \mathbf{A}_j) \quad (\text{donde } \nabla_i \cdot \mathbf{A}_j = 0) \\
 &= \sum_j m_j \mathbf{A}_j \cdot \nabla_i W_{ij}
 \end{aligned} \tag{2.9}$$

Análogamente tomamos el operador rotacional 2.6. El signo proviene de la anticonmutatividad del operador rotacional:

$$\nabla_i \times \mathbf{A}_i = - \sum_j m_j \mathbf{A}_j \times \nabla_i W_{ij} \tag{2.10}$$

Finalmente obtenemos las definiciones suavizadas de SPH para los operadores de A :

$$\nabla_i \mathbf{A}_i = - \frac{1}{\rho_i} \sum_j m_j \mathbf{A}_{ij} \cdot \nabla_i W_{ij} \tag{2.11}$$

$$\nabla_i \cdot \mathbf{A}_i = - \frac{1}{\rho_i} \sum_j m_j \mathbf{A}_{ij} \cdot \nabla_i W_{ij} \tag{2.12}$$

$$\nabla_i \times \mathbf{A}_i = \frac{1}{\rho_i} \sum_j m_j \mathbf{A}_{ij} \times \nabla_i W_{ij} \tag{2.13}$$

donde $A_{ij} = A_i - A_j$. Si A es constante la ecuación 2.3 es no nula, sin embargo la ecuación 2.11 si lo es.

2.1.1. Kernel

La función interpoladora o kernel es una función crucial para la evaluación de las ecuaciones del SPH. Se describe con $W(\mathbf{r}, h)$, donde $\mathbf{r} = \mathbf{r}_i - \mathbf{r}_j$ es la distancia entre la partícula i y cada una de sus vecinas j . Su valor se encuentra suavizado según la variable h_i . Gracias a él conseguimos la interpolación necesaria en las propiedades de las partículas vecinas a la que estamos tratando y, por tanto, la elección del kernel es crítica.

Para elegir un buen Kernel necesitamos que se cumpla las siguientes propiedades:

- Que cumpla la condición de normalizado.

$$\int W((\mathbf{r}' - \mathbf{r}, h) d\mathbf{r}' = 1 \tag{2.14}$$

- Que tenga simetría esférica. Esto es así para lograr una interpolación de segundo orden. Sin embargo, cuando aparecen fuertes impactos podría ser interesante deformar el núcleo con el fin de obtener una mejor resolución en el frente de choque. Esto se puede hacer cambiando la geometría del núcleo (*Owen y colaboradores, 1998* [69]), o construirlos de forma centralmente mas condensados (*Cabezón y colaboradores, 2008* [17]).
- Que reproduzca la función delta de Dirac con $h \rightarrow 0$.

$$\lim_{h \rightarrow 0} W(\mathbf{r}' - \mathbf{r}, h) = \delta(\mathbf{r}' - \mathbf{r}) \quad (2.15)$$

- Que sea una función picuda: Para que las partículas mas cercanas contribuyan en mayor medida a la interpolación que las más lejanas.
- Que tenga soporte compacto: Por encima de la distancia $2h_i$, la influencia de las partículas vecinas debe ser nula. Es decir, el número de contribuciones para calcular las propiedades de una partícula es finito, siendo el coste del orden de $\theta(n \cdot n_v)$, donde n el número de partículas y n_v el número de vecinos, en vez de $\theta(n^2)$, que sería el coste del cálculo de la función sin soporte compacto.

$$W((\mathbf{r}' - \mathbf{r} > 2h), h) = 0 \quad (2.16)$$

Esta condición es crucial para la versión implícita que queremos desarrollar, ya que influye fuertemente en la dispersión de la matriz a invertir, como se explicará más adelante.

El orden de interpolación se puede deducir de la evaluación del error cuando la función $A(\mathbf{r})$ se aproxima mediante el kernel. Tomando la expansión de la función $A(r)$ obtenemos lo siguiente:

$$A(r) \approx A(r_0) + \left(\frac{\partial A}{\partial r} \right)_{r_0} (r - r_0) + \frac{1}{2} \left(\frac{\partial^2 A}{\partial r^2} \right)_{r_0} (r - r_0)^2 + \dots \quad (2.17)$$

Y si calculamos este interpolante encontramos:

$$\langle A(r) \rangle = \int A(r) W(r - r_0, h) dr \approx A(r_0) \int W(r - r_0, h) dr + \quad (2.18)$$

$$+ \left(\frac{\partial A}{\partial r} \right)_{r_0} \int (r - r_0) W(r - r_0, h) dr + \quad (2.19)$$

$$+ \frac{1}{2} \left(\frac{\partial^2 A}{\partial r^2} \right)_{r_0} \int (r - r_0)^2 W(r - r_0, h) dr + \dots = A(r_0) + O(h^2) \quad (2.20)$$

En el primer término tenemos que la integral es la condición de normalización y por tanto su valor es 1. En el segundo término tenemos que la integral es nula, si el kernel es una función par. En el tercer término tenemos la segunda derivada del kernel por lo que la precisión es exacta si la función es lineal.

Simetrización del valor del kernel La función kernel no es en la práctica totalmente simétrica entre partículas ya que cada una de ellas tiene su propia h . Incluso puede suceder que la partícula j sea vecina de i y sin embargo que i no sea vecina de j . Algunas versiones del método SPH exigen simetría en el kernel. En el código aquí presentado, la física adicional que se agrega a las ecuaciones básicas de Euler necesitan este tipo de simetrización en el valor del kernel (gravedad, viscosidad artificial y conducción térmica). Con este sistema se consigue asegurar la reciprocidad de los pares de vecinos.

Existen dos formas de conseguirlo. La primera consiste en simetrizar la longitud de suavizado, de tal forma que el kernel resultante tiene la h modificada. Es la forma mas simple y el valor de la longitud de suavizado es la media aritmética de los valores de h_i y h_j .

$$W_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_{ij}) \quad ; \quad h_{ij} = \frac{h_i + h_j}{2} \quad (2.21)$$

La segunda consiste en obtener el valor medio de los kernels calculados con la longitud de suavizado de i y de j .

$$\tilde{W}_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_i, h_j) = \frac{W_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_i) + W_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_j)}{2} \quad (2.22)$$

En el código se ha utilizado la ecuación 2.22 para describir la simetría del valor del kernel (vease Sec. 2.2.4, 2.2.5 y 2.2.6). Aunque en principio ambas descripciones son equivalentes la segunda opción garantiza que cada partícula tenga una longitud de suavizado bien definida.

Kernel Cubic Spline En las primeras versiones, el código se utilizaba de forma general el kernel *cubic spline* que fue propuesto por Monaghan y Lattanzio, 1985 [63] y está basado en las funciones M_4 B-spline de Schoenberg y Sharma, 1973 [88].

$$W(v, h) = C \cdot \begin{cases} 1 - \frac{3}{2}v^2 + \frac{3}{4}v^3, & (0 \leq v < 1) \\ \frac{1}{4}(2-v)^3, & (1 \leq v < 2) \\ 0, & (v \geq 2) \end{cases} \quad (2.23)$$

$$C = \frac{1}{K h^d} ; \quad K = \left(\frac{2}{3}, 1D\right), \left(\frac{10}{7 \cdot \pi}, 2D\right), \left(\frac{1}{\pi}, 3D\right) ; \quad v = \frac{\mathbf{r}_{ij}}{h} \quad (2.24)$$

Siendo v la proporción de la distancia entre partículas y la longitud de suavizado y C la constante de normalización y donde K depende de las dimensiones d . El detalle de las derivadas del kernel se puede encontrar en el apéndice A.1.

Kernel Sinc Recientemente se han propuesto una nueva familia de kernels adaptativos más precisos (*Cabezón y colaboradores, 2008* [17]), basados en funciones armónicas y que cubre toda la gama de núcleos spline de manera continua. Estos kernels tienen formas mas condensadas o mas extendidas y se ajustan con un parámetro n con el fin de aumentar la resolución local cuando es necesario (por ejemplo, un valor $n = 4.89$ obtiene un valor similar al del kernel quíntico).

$$W_n^S(v, h, n) = \frac{B_n}{h^d} S_n\left(\frac{\pi}{2}v\right) \quad (0 \leq v \leq 2) \quad (2.25)$$

$$B_n = \begin{cases} b_0 + b_1 n^{1/2} + b_2 n + b_3 n^{-1/2}, & 1D \\ b_0 + b_1 n + b_2 n^{-1} + b_3 n^{-2}, & 2D \\ b_0 + b_1 n^{1/2} + b_2 n + b_3 n^{3/2}, & 3D \end{cases} \quad (2.26)$$

Donde:

- b_0, b_1, b_2 y b_3 son constantes y dependen del número de dimensiones (valores descritos en el apéndice A.2).
- $v = (r_{ij}/h)$, n (índice del kernel), B_n (constante de normalización), h (longitud de suavizado) y d (número de dimensiones).
- $S_n\left(\frac{\pi}{2}v\right) = \text{sinc}^n\left(\frac{\pi}{2}v\right)$
- $\text{sinc}\left(\frac{\pi}{2}v\right) = \left[\frac{\sin\left(\frac{\pi}{2}v\right)}{\left(\frac{\pi}{2}v\right)} \right]$

Una de las ventajas principales de usar el kernel Sinc es que reduce la aparición del “pairing-instability” dentro de la simulación.

En la última versión, el código utiliza de forma general el kernel Sinc para todos los cálculos y se ha dejado como opciones el uso del kernel Cubic Spline y el Kernel Gaussiano. El detalle de la implementación de los kernels del código se puede encontrar descrita en el apéndice A.

2.1.2. Longitud de suavizado

El parámetro h_i es una de las magnitudes mas importantes del método SPH. Esta variable determina la longitud de suavizado del kernel y establece la distancia a la cual las partículas

vecinas contribuyen a las propiedades de la partícula. Si el kernel tiene alcance compacto y este viene determinado por el parámetro h , se suele tomar como valor de alcance $2 h_i$ (Fig. 2.1).

Como el estado del sistema cambia con el tiempo, el número de vecinos también varía y por tanto la longitud de suavizado debe variar con la distancia y el tiempo $h(r, t)$. Si no se modifica el parámetro podría pasar cosas raras, es decir por ejemplo que una partícula se aislara y dejara de ver vecinos y estos no contribuirían a la variación de sus propiedades; o por contra si una partícula se encontrará en una zona donde aumenta la densidad, se produciría un aumento muy significativo del número de vecinos produciendo un exceso de suavizado y un aumento innecesario del coste computacional. La variación de h evita estos problemas, de tal modo que mantiene relativamente constante el número de vecinos, ajustándolo a un parámetro definido por el usuario y que establece el número de vecinos objetivo para cada partícula.

Para actualizar la h_i se puede utilizar la definición siguiente (*Serichol, 2005* [90]).

$$h_i = h_0 \frac{1}{2} \left(1 + (2^{\eta_h} - 1) \frac{n_{v_\phi}}{n_{v_0}} \right)^{1/\eta_h} \quad (2.27)$$

donde h_0 es la longitud de suavizado actual, n_{v_0} es el número de vecinos actual, n_{v_ϕ} el número de vecinos objetivo o ideal y η_h es un parámetro de control que especifica la velocidad de ajuste de la h_i al n_{v_ϕ} .

El valor de n_{v_ϕ} se especifica diferente según las dimensiones pero se le suelen asignar valores en el rango de 25 a 100 vecinos. Los modelos físicos necesitan mayor número de n_{v_ϕ} a mayor número de dimensiones utilizadas. También hay que indicar que la resolución y el coste computacional están directamente relacionados con el valor del parámetro. Por tanto el objetivo es encontrar un compromiso entre el número de partículas, la resolución requerida y el tiempo de cálculo asumible. Nuestra experiencia dice que los modelos en 3D con una cifra a partir de 50 vecinos dan resultados con una resolución aceptable.

A η_h se le suele dar un valor en el rango de 2 a 10. Con valores bajos obtenemos ajustes mas rápidos, con el inconveniente de que si la red de partículas se encuentra muy ordenada en sucesivas iteraciones h_i tiene pequeñas oscilaciones respecto del valor teórico. Si el problema tratado es sensible a estas pequeñas oscilaciones es necesario aumentar η_h para que se minimice este efecto.

En la formulación variacional la evolución de h se calcula como $h \propto (1/\rho)^{(1/d)}$, donde d es el número de dimensiones.

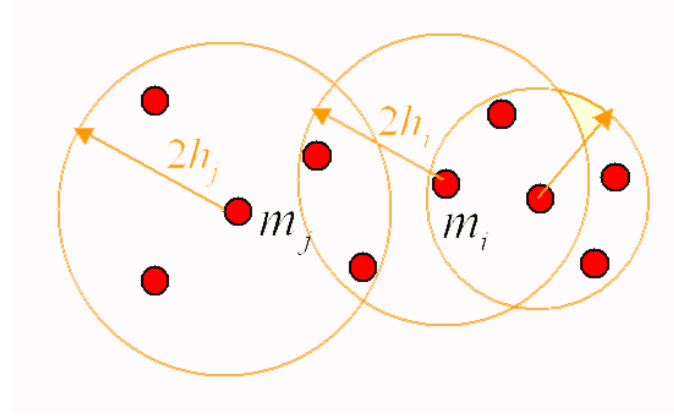


Figura 2.1: Representación esquemática de las partículas m_i , m_j y sus vecinas. Cada partícula tiene el mismo número de vecinos por tanto cuando la densidad es mayor el radio $2h$ es menor.

2.1.3. Integración y evolución temporal

Habitualmente en SPH el método de integración es de tipo explícito. Es decir los valores de las variables de las partículas a un tiempo determinado ($i + 1$), se establecen a partir de los valores de las variables conocidas en el paso de tiempo anterior (i). La gran limitación de los métodos explícitos se encuentra en la condición de Courant (*Courant y colaboradores, 1967* [22]). Esto produce que el paso de tiempo máximo adecuado sea siempre muy pequeño y reduce la aplicabilidad a escenarios muy dinámicos. En contraposición a los códigos SPH explícitos este documento introduce una versión implícita que por construcción no tiene esta limitación en el paso de tiempo, pero a cambio el coste computacional es mayor. El principal problema es que requiere la inversión de una matriz $m \times m$, donde $m = n * v$, siendo n el número de partículas y v el número de variables independientes por partícula.

El esquema de resolución utilizado en el código que aquí se presenta tiene dos posibilidades en cuanto a su precisión. Dispone de un parametro, θ en las ecuaciones (véase sección 2.3.2). Este parámetro establece una ponderación en las propiedades según su valor en la iteración anterior (forma explícita) y en la iteración actual (forma implícita). Pudiendo seleccionar, según el valor de θ , entre un esquema totalmente explícito $\theta = 0$ (a primer orden), totalmente implícito $\theta = 1$ (a primer orden), o semi-implícito $\theta = 0.5$ (centrado, a segundo orden).

Cabe destacar que, en el código desarrollado, la ejecución en forma explícita solo se utiliza para hacer pruebas comparativas usando el mismo código. Esto es debido a que el programa se ha desarrollado pensando en la ejecución implícita y semi-implícita, de tal forma que en la ejecución explícita el código es subóptimo por construcción. Si es necesario utilizar entornos de producción con SPH explícito es una buena idea utilizar códigos optimizados en esta forma, como por ejemplo el resto de códigos SPH desarrollados en nuestro grupo de investigación. Como

el objetivo es verificar el método, los test que se describen en este documento se han calculado con $\theta = 1$.

El paso de tiempo se actualiza una vez obtenemos la convergencia del esquema *Newton – Raphson* (véase sección 2.3.4) utilizado para el cálculo de las ecuaciones. Para ello se tienen en cuenta una serie de criterios (véase sección 3.2.2). No todos los criterios se utilizan simultaneamente, depende del escenario concreto a simular se activan o desactivan.

2.2. Ecuaciones

La técnica SPH fue desarrollada originalmente en coordenadas cartesianas en 3D, esta sección es un resumen de los métodos más comunes que se utilizan para construir versiones suavizadas de las ecuaciones de los fluidos de Euler (Cuadro 2.3), para evaluar la física adicional necesaria para implementar los modelos y acomodarlos a la realidad observada (disipación de energía, gravedad, conducción térmica, ...).

También hay que indicar la existencia de nuevos trabajos sobre la base matemática del método (*García-Senz y colaboradores, 2012* [26]). En ellos se aportan nuevas visiones de la técnica SPH que consiguen una mejora apreciable en la estimación de los gradientes y por tanto mejoran la calidad de las simulaciones (véase, por ejemplo, *Hopkins, 2015* [41], donde se hace una comparativa exhaustiva de las diferentes modificaciones del SPH).

Si tomamos la ecuación básica de cualquier propiedad en SPH (ecuación 2.2) podemos destacar los siguientes detalles importantes del método en referencia a las ecuaciones. Según se especifica en *Monaghan, 2005* [61], una de las mejores características de la formulación SPH es la simplicidad para el cálculo de las derivadas espaciales de cualquier magnitud A_i , donde χ_i equivale a cualquier dimensión. Esto es debido a que el kernel es una función diferenciable exactamente.

$$\frac{\partial A_i}{\partial \chi_i} = \sum_{j=1}^N m_j \frac{A_j}{\rho_j} \frac{\partial W_{ij}}{\partial \chi_i} \quad (2.28)$$

Como alternativa, en la literatura podemos encontrar diferentes formas de esta ecuación:

$$\frac{\partial A_i}{\partial \chi_i} = \sum_{j=1}^N \frac{m_j}{\rho_j} (A_j - A_i) \frac{\partial W_{ij}}{\partial \chi_i} \quad (2.29)$$

$$\frac{\partial A_i}{\partial \chi_i} = \frac{1}{\rho_i} \sum_{j=1}^N m_j (A_j - A_i) \frac{\partial W_{ij}}{\partial \chi_i} \quad (2.30)$$

En las próximas secciones observaremos como se aplica esta propiedad a las diferentes ecua-

ciones de la hidrodinámica de Euler y así obtener las ecuaciones que se utilizarán en el SPH.

2.2.1. Aproximación de las derivadas en las ecuaciones diferenciales

Las derivadas son aproximadas por cocientes incrementales, es decir, la ecuación diferencial pasa a ser una ecuación algebraica. En las siguientes ecuaciones podemos observar aproximaciones a una función cualquiera $f(x)$.

Aproximación de la primera derivada de $f(x)$ Aquí se exponen tres métodos de aproximación a la derivada, difiriendo entre ellos en el orden de convergencia obtenido. La aproximación será mejor cuando mayor sea el orden de convergencia.

Aproximación <i>backward</i> (1 ^{er} orden):	$\frac{\partial f(x_o)}{\partial x} = \frac{f(x_o) - f(x_o - h)}{h} + O(h) \quad (2.31)$
Aproximación <i>forward</i> (1 ^{er} orden):	$\frac{\partial f(x_o)}{\partial x} = \frac{f(x_o + h) - f(x_o)}{h} + O(h) \quad (2.32)$
Aproximación <i>centrada</i> (2 ^o orden):	$\frac{\partial f(x_o)}{\partial x} = \frac{f(x_o + h) - f(x_o - h)}{2 h} + O(h^2) \quad (2.33)$

Cuadro 2.1: Aproximación de la primera derivada de $f(x)$.

Aproximación de la segunda derivada de $f(x)$ Se muestra directamente la aproximación centrada a segundo orden.

Serie de Taylor Dada una función f lo suficientemente diferenciable, si aplicamos un desarrollo en forma de serie de Taylor obtenemos la ecuación 2.35.

$$f(x_o + h) = f(x_o) + \frac{1}{1!} \frac{\partial f(x_o)}{\partial x} h + \frac{1}{2!} \frac{\partial^2 f(x_o)}{\partial x^2} h^2 + \dots + \frac{1}{n!} \frac{\partial^n f(x_o)}{\partial x^n} h^n + \varepsilon \quad (2.35)$$

Aproximación <i>centrada</i> (2º orden):	$\frac{\partial^2 f(x_o)}{\partial x^2} = \frac{f(x_o - h) - 2f(x_o) + f(x_o + h)}{h^2} + O(h^2) \quad (2.34)$
--	--

Cuadro 2.2: Aproximación de la segunda derivada de $f(x)$.

donde el error se define en la ecuación 2.36 y esta determinado por el parámetro α situado en el rango $x_o \leq \alpha \leq x_o + h$.

$$\varepsilon = \frac{1}{(n+1)!} \frac{\partial^{n+1} f(x_o)}{\partial x^{n+1}} h^{n+1} \quad (2.36)$$

2.2.2. Hidrodinámica de Euler: SPH estándar

Las partículas de los modelos se comportan habitualmente como fluidos y por tanto se puede describir su evolución mediante las ecuaciones hidrodinámicas de Euler. Estas ecuaciones determinan la variación con respecto al tiempo en cada partícula de su posición (r), velocidad (v), energía interna (U), temperatura (T) y densidad (ρ) (Cuadro 2.3).

La ecuación de la energía 2.39 se obtiene de la primera ley de la termodinámica para un sistema aislado (Rosswog, 2009 [78]):

$$T ds = dU - \frac{P}{\rho^2} d\rho \quad (2.42)$$

La relación entre temperatura y energía interna 2.40 es exacta en gases ideales, en el caso más general corresponde a la ecuación siguiente.

$$dU = C_v dT + \left[T \left(\frac{\partial P}{\partial T} \right)_V - P \right] dV \quad (2.43)$$

Ecuación del movimiento Se puede utilizar directamente la ecuación 2.37 o introducir una corrección denominada XSPH, introducida por Monaghan, 1989 [57] y detallada en Monaghan, 1992 [58].

$$\frac{d\mathbf{r}_i}{dt} = \hat{\mathbf{v}}_i = \mathbf{v}_i + \varepsilon \sum_{j=1}^N m_j \frac{(\mathbf{v}_i - \mathbf{v}_j)}{\rho_{ij}} W_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_i) \quad (2.44)$$

donde $\rho_{ij} = (\rho_i + \rho_j)/2$ y $0 \leq \varepsilon \leq 1$ es una constante. La variante XSPH mueve la partícula con una velocidad que es más cercana a la velocidad media de su entorno. Se debe tener en cuenta que

Movimiento:	$\frac{d\mathbf{r}}{dt} = \frac{\mathbf{v} + \mathbf{v}^{(n-1)}}{2} \quad (2.37)$
Momento:	$\frac{d\mathbf{v}}{dt} = \dot{\mathbf{v}} = \mathbf{a} = -\frac{\nabla P}{\rho} + \mathbf{g} \quad (2.38)$
Energía interna:	$\frac{dU}{dt} = \dot{U} = -\frac{P}{\rho} \nabla \cdot \mathbf{v} \quad (2.39)$
Temperatura:	$\frac{dT}{dt} = \dot{T} = \frac{1}{C_v} \cdot \frac{dU}{dt} \quad (2.40)$
Continuidad:	$\frac{d\rho}{dt} = \dot{\rho} = -\rho \nabla \cdot \mathbf{v} \quad (2.41)$

Cuadro 2.3: Ecuaciones hidrodinámicas de Euler. Donde \mathbf{r} es el vector posición, \mathbf{v} vector velocidad, P presión, g aceleración gravitatoria, U energía interna por gramo, ρ densidad, T temperatura y C_v calor específico a volumen constante.

esta modificación no introduce disipación pero incrementa la dispersión. Esta corrección de la velocidad mantiene las partículas más ordenadas, y, para velocidades de flujo altas y contornos, evita la penetración del fluido en el contorno.

Si utilizamos la ecuación de continuidad (ecuación 2.41) es necesario aplicar la corrección para mantener una estricta consistencia. Es decir hay que utilizar $\hat{\mathbf{v}}$ en vez de \mathbf{v} .

Aunque la opción XSPH está implementado los test presentados en el documento utilizan $\varepsilon = 0$, es decir, usan la ecuación 2.37.

Ecuación del momento Trata la aceleración de las partículas. En esta descripción de la adaptación de la ecuación del momento al SPH se obviará la gravedad ya que se trata extensamente más adelante (véase Sec. 2.2.4). Para cumplir con las leyes de conservación es necesario simetrizar la ecuación 2.38,

$$\frac{\nabla P}{\rho} = \nabla \left(\frac{P}{\rho} \right) + \frac{P}{\rho^2} \nabla \rho \quad (2.45)$$

Aplicando la relación 2.3 y utilizando la ecuación anterior 2.45 obtendremos la ecuación suavizada del momento.

$$\frac{d\mathbf{v}_i}{dt} = - \sum_{j=1}^N m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} \quad (2.46)$$

Ecuación de la energía Si aplicamos la relación 2.12 a la ecuación 2.39 podemos obtener la ecuación suavizada de la energía.

$$\frac{dU_i}{dt} = \frac{P_i}{\rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.47)$$

A pesar de que la ecuación 2.47 es la que se usa habitualmente para la energía, tiene un problema con la simetría. Para evitarlo podemos desarrollar la ecuación 2.39 para que sea simétrica.

$$\frac{P}{\rho} \nabla \cdot \mathbf{v} = \nabla \cdot \left(\frac{P}{\rho} \mathbf{v} \right) - \mathbf{v} \cdot \nabla \left(\frac{P}{\rho} \right) \quad (2.48)$$

Aplicando la relación anterior 2.12, a la nueva ecuación 2.48 obtendremos la nueva ecuación suavizada de la energía.

$$\frac{dU_i}{dt} = \sum_{j=1}^N m_j \frac{P_j}{\rho_j^2} (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.49)$$

Promediando las ecuaciones 2.47 y 2.49 obtenemos la versión simétrica de la ecuación de la energía.

$$\frac{dU_i}{dt} = \frac{1}{2} \sum_{j=1}^N m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.50)$$

La ecuación 2.50 tiene el mismo tipo de simetría que la ecuación del momento 2.46. De todos modos la elección de las ecuaciones 2.47 o 2.49, en general, produce resultado similares, aunque la ecuación 2.50 debe ser solo utilizada en situaciones donde la magnitud P/ρ^2 no cambia mucho a lo largo de la longitud de suavizado (h).

Ecuación de la temperatura Si incluimos en el esquema la variación de la temperatura, en vez de la variación de energía interna, podemos utilizar la relación termodinámica:

$$P = T \left(\frac{\partial P}{\partial T} \right)_\rho + \rho^2 \left(\frac{\partial U}{\partial \rho} \right)_T \quad (2.51)$$

para obtener la ecuación 2.53 derivada de la relación entre ellas (ecuación 2.52) y la ecuación 2.47.

$$\frac{\partial T}{\partial t} = \frac{1}{C_v} \cdot \frac{\partial U}{\partial t} \quad (2.52)$$

$$\frac{dT}{dt} = \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \frac{1}{\rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.53)$$

Esta ecuación es por ejemplo utilizada en *Lorén-Aguilar, 2009* [52] para simular objetos compactos.

Ecuación de la continuidad Si aplicamos la relación 2.12 a la ecuación 2.41 se obtiene la ecuación suavizada de la continuidad:

$$\frac{d\rho_i}{dt} = \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.54)$$

Esta forma de la ecuación de la continuidad tiene la ventaja de que es exactamente igual a 0 cuando la velocidad es constante, es decir, como era de esperar la densidad no varía. Por contra, aunque la ecuación 2.54 permite calcular el ratio de variación de la densidad en SPH, no conserva la masa exactamente. A pesar de ello la desviación es pequeña y no representa variaciones apreciables para nuestros cálculos.

Además, en SPH, podemos obtener la densidad de una forma particularmente simple y que conserva la masa exactamente. Para ello tomamos la ecuación 2.2 y sustituimos la propiedad $A_i(\mathbf{r})$ por la densidad de la partícula i (ρ_i) y A_j por la densidad de la partícula j (ρ_j).

$$\rho_i = \sum_{j=1}^N m_j W_{ij}(|\mathbf{r}_i - \mathbf{r}_j|, h_i) \quad (2.55)$$

El código que aquí se presenta tiene implementadas las dos formas de tratamiento de la densidad, pero de forma general se utilizará la ecuación 2.55, porque es compatible con la formulación lagrangiana del SPH y además computacionalmente mas eficiente, Con ella es necesario recalcular la densidad a cada paso de tiempo a partir de las posiciones, pero a cambio no requiere su inclusión en el sistema de ecuaciones que luego hay que invertir (véase Sec. 2.3.4). La inversión de la matriz tiene un coste algorítmico superior al recalclo de densidad que tiene un coste de $\theta(n \cdot \log n)$.

Resumen de las ecuaciones SPH de la hidrodinámica de Euler El cuadro 2.4 contiene las ecuaciones estándar que se utilizan en el SPH.

Movimiento:	$\frac{d\mathbf{r}_i}{dt} = \frac{\mathbf{v}_i + \mathbf{v}_i^{(n-1)}}{2} + \varepsilon \sum_{j=1}^N m_j \frac{(\mathbf{v}_i - \mathbf{v}_j)}{\rho_{ij}} W_{ij}(\mathbf{r}_i - \mathbf{r}_j , h_i) \quad (2.56)$
Momento:	$\frac{d\mathbf{v}_i}{dt} = - \sum_{j=1}^N m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_i W_{ij} + \mathbf{g}_i \quad (2.57)$
Energía interna:	$\frac{dU_i}{dt} = \frac{P_i}{\rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.58)$
Temperatura:	$\frac{dT}{dt} = \left(\frac{\partial P_i}{\partial T_i} \right)_\rho \cdot \frac{T_i}{C_v} \cdot \frac{1}{\rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.59)$
Continuidad:	$\frac{d\rho_i}{dt} = \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (2.60)$
Densidad:	$\rho_i = \sum_{j=1}^N m_j W_{ij}(\mathbf{r}_i - \mathbf{r}_j , h_i) \quad (2.61)$
Distancia de interpolación:	$h_i = \left(\frac{K}{\rho} \right)^{1/d}, \quad h_i = h_0 \frac{1}{2} \left(1 + (2^{\eta_h} - 1) \frac{n_{v\phi}}{n_{v0}} \right)^{1/\eta_h} \quad (2.62)$

Cuadro 2.4: Ecuaciones hidrodinámicas de Euler en SPH. Donde \mathbf{r} es el vector posición, \mathbf{v} es el vector velocidad, P es la presión, g es la aceleración gravitatoria, U es la energía interna por gramo, ρ es la densidad, T es la temperatura y C_v es el calor específico a volumen constante.

2.2.3. Formalismo Euler-Lagrange: SPH variacional

En la aproximación variacional las partículas del sistema están vinculadas a una serie de ecuaciones evolutivas derivadas del formalismo Euler-Lagrange, descrito en *Monaghan, 2005* [61]. En él se detalla que mediante el Lagrangiano, las ecuaciones se pueden describir de forma natural

con un esquema simetrizado e incluyendo los gradientes espaciales de la longitud de suavizado. Se basa en la ecuación de Lagrange para fluidos en movimiento en sistemas no disipativos, descrita por *Eckart* en 1960. La forma SPH para la ecuación de Lagrange es la siguiente.

$$L = \sum_{i=1}^N m_i \left(\frac{1}{2} v_i^2 - U_i(\rho_i, s_i) - \Phi_i \right) \quad (2.63)$$

donde m es la masa, v es la velocidad, U la energía interna por gramo, ρ la densidad, s la entropía y Φ el potencial gravitatorio por unidad de masa.

Según se describe en *Springel, 2010* [95] en SPH podemos discretizar el Lagrangiano en términos de fluidos de partículas de masa m_i .

$$L_{SPH} = \sum_{i=1}^n \left(\frac{1}{2} m_i v_i^2 - m_i U_i \right) \quad (2.64)$$

La definición de la ecuación de Lagrange para una partícula concreta i .

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \mathbf{v}_i} \right) - \frac{\partial L}{\partial \mathbf{r}_i} = 0 \quad (2.65)$$

El momento es dado directamente a partir de 2.63.

$$\frac{\partial L}{\partial \mathbf{v}_i} = m_i \mathbf{v}_i \quad (2.66)$$

Las derivadas espaciales en el Lagrangiano se encuentran asumiendo que la entropía es constante y por lo tanto la energía interna puede ser expresada como una función de la densidad.

$$\frac{\partial L}{\partial \mathbf{r}_i} = - \sum_{j=1}^N m_j \left(\frac{\partial U_j}{\partial \rho_j} \right)_s \frac{\partial \rho_j}{\partial \mathbf{r}_i} \quad (2.67)$$

donde la derivada de la energía respecto de la densidad es dada por la primera ley de la termodinámica a entropía constante, $dU = -PdV$, donde $V = m/\rho$ es el volumen de la partícula cuando la variación en energía por unidad de masa viene dado por:

$$dU = \frac{P}{\rho^2} d\rho \quad (2.68)$$

La longitud de suavizado es una función de la densidad. El método usual que se presenta en la literatura es o bien calcular la longitud de suavizado utilizando el valor actual de la densidad (calculada con la ecuación 2.55), o calcularla para la densidad de acuerdo con la expresión siguiente.

$$\frac{d \ln h}{dt} = - \frac{1}{d} \frac{d \ln \rho}{dt} \quad (2.69)$$

Entonces, varias técnicas pueden ser usadas para ajustar el valor de h . Por ejemplo, el promedio de la densidad local es utilizado para cambiar h en *Steinmetz y Mueller, 1993* [97]; Otro método utilizado es ajustar el modelo de manera que cada partícula tenga un número constante de los vecinos como en *Hernquist y Katz, 1989* [38]. Idealmente, debería determinarse a partir de la ecuación 2.54 de modo que sea consistente con la densidad obtenida con la ecuación 2.55. Además, a veces es necesario establecer un límite máximo en h_i cuando alguna partícula tiene una densidad es muy pequeña ρ_i , y una h_i muy grande, con la intención de evitar interacciones fuertes con partículas con alta densidad, tal que $\rho_i \ll \rho_j$.

Esto puede lograrse si la expresión de h se sustituye según se describe en *Monaghan, 2005* [61].

$$h_i = \eta_h \left(\frac{m_i}{\rho_i} \right)^{1/d} \quad (2.70)$$

donde η_h es un parámetro que suele tomar un valor en el rango [1.2, 1.3] y d es el número de dimensiones.

La forma ideal es, no obstante, introducir h_i como una ligadura que expresa la conservación de la masa $C = \rho_i h_i^d$ e incorporarla a las ecuaciones a través del parámetro Ω (*Springel y Hernquist, 2002* [96]).

Parámetro Ω_i Según se describe en la ecuación 2.55 la variable ρ_i depende de h_i y viceversa. Si consideramos la variación de la h , la derivada temporal de la densidad viene dada por:

$$\begin{aligned} \frac{d\rho_i}{dt} &= \frac{d}{dt} \left(\sum_{j=1}^N m_j W_{ij}(h_i) \right) \\ &= \sum_{j=1}^N m_j \left(\frac{\partial W_{ij}(h_i)}{\partial r_{ij}} \frac{dr_{ij}}{dt} + \frac{\partial W_{ij}(h_i)}{\partial h_i} \frac{dh_i}{dt} \right) \\ &= \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial r_{ij}} \hat{\mathbf{e}}_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) + \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h_i} \cdot \frac{\partial h_i}{\partial \rho_i} \frac{d\rho_i}{dt} \\ &= \sum_{j=1}^N m_j \mathbf{v}_{ij} \cdot \nabla_i W_{ij}(h_i) + \frac{\partial h_i}{\partial \rho_i} \frac{d\rho_i}{dt} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h_i} \end{aligned} \quad (2.71)$$

Observando esta derivada podemos introducir el parámetro Ω_i , que contiene el gradiente de h_i , si le damos el siguiente valor:

$$\Omega_i = 1 - \frac{\partial h_i}{\partial \rho_i} \sum_{j=1}^n m_j \frac{\partial W_{ij}(h_i)}{\partial h_i} \quad (2.72)$$

De tal forma que la derivada temporal de la densidad queda así:

$$\frac{d\rho_i}{dt} = \frac{1}{\Omega_i} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij}(h_i) \quad (2.73)$$

De la misma forma consideramos la derivada espacial de la densidad:

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \sum_{k=1}^N m_k \left(\nabla_i W_{jk}(h_j) + \frac{\partial W_{jk}(h_j)}{\partial h_j} \frac{\partial h_j}{\partial \rho_j} \frac{\partial \rho_j}{\partial \mathbf{r}_i} \right) = \frac{1}{\Omega_i} \sum_{j=1}^N m_j \nabla_i W_{ij}(h_i) \quad (2.74)$$

Cuando se contemplan variaciones en la h (gradientes de h) se aumenta la precisión del SPH y de las propiedades de conservación (especialmente del binomio energía-entropía). La importancia de la contribución de este término depende del problema concreto y de la resolución numérica que se utilice.

Resumiendo, si las derivadas de la longitud de suavizado (gradientes de h) se tienen en cuenta, el SPH estándar ha de ser corregido por el factor $1/\Omega$ (ecuaciones 2.72, 2.73, 2.74). A partir de estas ecuaciones, según se describe detalladamente en *Rosswog, 2009* [78], se pueden obtener las nuevas ecuaciones para el momento, energía y temperatura (Cuadro 2.6). Las ecuaciones obtenidas conservan el momento y, por construcción, la energía, masa y entropía.

La última versión del código y todos los test mostrados en este documento incluyen la opción lagrangiana con el parámetro Ω , que la opción considerada más actual y considerada más precisa en la definición de las ecuaciones del SPH (y que es utilizada en los test presentados más adelante).

Conservación del momento y la energía Según se explica en *Monaghan, 2005* [61], las leyes de conservación se pueden deducir ya sea de las ecuaciones del movimiento o de la invariancia de las transformaciones infinitesimales de Lagrange. El método SPH, imita un sistema de partículas con fuerzas entre ellas, pero con diferente grado de interacción a través de la presión, la densidad y tiene dependencia geométrica a través del kernel (depende de las posiciones de las partículas).

El momento lineal y angular son conservados siempre que el Lagrangiano (ecuación 2.63) sea invariante a traslaciones y rotaciones. Como en SPH la densidad y la energía interna, a entropía constante, son invariantes a traslaciones y rotaciones, los términos de la dinámica de fluidos conservan el momento lineal y angular por construcción. Si tomamos de la ecuación del momento 2.80 lo siguiente:

$$\nabla_i W_{ij}(h_i) = \mathbf{r}_{ij} F_{ib}(h_i) \quad (2.75)$$

donde $F_{ij}(h_i) = F(|\mathbf{r}_{ij}|, h_i)$, por simetría de los términos de interacción, el momento lineal y angular son conservados exactamente. Además, como no hay una dependencia explícita del tiempo en el Lagrangiano, la energía también es conservada.

Para comprobar la conservación de la energía a cada paso de tiempo se pueden observar la variación de las ecuaciones del cuadro 2.5

Energía interna (U):	$U = \sum_{i=1}^N m_i \cdot u_i \quad (2.76)$
Energía cinética (E_k):	$E_k = \sum_{i=1}^N \frac{1}{2} m_i \cdot v_i^2 \quad (2.77)$
Energía potencial gravitatoria (E_g):	$E_g = -\frac{G}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{m_i m_j}{ r_{ij} } \quad (2.78)$
Energía total (E):	$E = U + E_k + E_g \quad (2.79)$

Cuadro 2.5: Cálculo de la conservación de la energía.

Resumen de las ecuaciones SPH del formalismo Euler-Lagrange El cuadro 2.6 contiene las ecuaciones variacionales que se utilizan en el SPH.

2.2.4. Auto-gravedad

Aunque SPH funciona mediante la interpolación de magnitudes de los vecinos cercanos, en los modelos astrofísicos es necesario introducir la auto-gravedad (*Springel, 2010* [95]), a diferencia de otras áreas de la mecánica de fluidos donde solo tienen influencia los campos gravitatorios externos. En SPH la auto-gravedad se puede agregar de forma sencilla y precisa, conservando la energía total, en cambio en los métodos de mallas no se conserva sin introducir en el código artificios adicionales (*Springel, 2010* [94])

Si representamos la energía potencial gravitatoria como Φ , la fuerza gravitacional \mathbf{f}_g es el

Momento:	$\frac{d\mathbf{v}_i}{dt} = - \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \nabla_i W_{ij}(h_i) + \frac{P_j}{\Omega_j \rho_j^2} \nabla_i W_{ij}(h_j) \right) + \mathbf{g}_i \quad (2.80)$
Energía:	$\frac{dU_i}{dt} = \frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij}(h_i) \quad (2.81)$
Temperatura:	$\frac{dT_i}{dt} = \left(\frac{\partial P_i}{\partial T_i} \right)_\rho \cdot \frac{T_i}{C_v} \cdot \frac{1}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij}(h_i) \quad (2.82)$
Continuidad:	$\frac{d\rho_i}{dt} = \frac{1}{\Omega_i} \sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij}(h_i) \quad (2.83)$

Cuadro 2.6: Ecuaciones del formalismo Euler-Lagrange.

negativo del gradiente de la energía potencial gravitatoria (ecuación 2.84). El potencial gravitatorio se identifica con el parámetro ϕ (ecuación 2.86), el campo gravitatorio o aceleración gravitatoria es el gradiente del potencial gravitatorio $\nabla\phi$ y se define como la fuerza gravitatoria por unidad de masa (ecuación 2.85).

$$\mathbf{f}_g = -\nabla\Phi \quad (2.84)$$

$$\nabla\phi \equiv -\frac{\nabla\Phi}{m} \quad (2.85)$$

$$\phi(\mathbf{r}) = G \int_{V'} \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} dV' \quad (2.86)$$

La gravedad de Newton determina la fuerza gravitatoria total que actúa sobre una partícula i de masa m y dice que la fuerza gravitatoria entre dos objetos es proporcional al producto de sus masas e inversamente proporcional al cuadrado de la distancia que las separa (ecuación 2.87). Tiene un radio de acción de largo alcance y para su cálculo, en SPH, se han de agregar

también las contribuciones de las partículas que no son vecinas, es decir el resto de partículas que se encuentran a una distancia mayor que la longitud de suavizado.

$$\mathbf{f}_{g_i} = -G m_i \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ij}^2} \hat{\mathbf{r}}_{ij} = -G m_i \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ij}^3} \mathbf{r}_{ij} \quad (2.87)$$

$$\nabla \phi_i = G \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ij}^3} \mathbf{r}_{ij} \quad (2.88)$$

$$\phi_i = -G \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ij}} \quad (2.89)$$

Gravedad exacta calculada de forma directa

Para resolver de forma exacta la ley de gravedad de Newton se tiene en cuenta todas las partículas del modelo y se calcula incluyendo \mathbf{g}_i (ecuación 2.57) directamente en la discretización SPH de la ecuación del momento (ecuaciones 2.132). Para ello se tiene que evaluar un conjunto de $n \cdot (n - 1)$ ecuaciones, el coste computacional de este cálculo es cuadrático $\theta(n^2)$, donde n es el número de partículas.

$$\mathbf{g}_i = -\nabla_i \phi_i = -G \sum_{j=1, j \neq i}^n \frac{m_j}{r_{ij}^3} \mathbf{r}_{ij} \quad (2.90)$$

Gravedad aproximada calculada con la ley de Gauss

La ley de Gauss, aplicada a la gravedad, es una ley de la física esencialmente equivalente a la ley de gravedad de Newton. Tal y como se describe en *García-Senz y colaboradores, 2009* [27], es utilizada para simplificar el cálculo de la gravedad en situaciones específicas con alta simetría (punto, línea, plano, cilindro, esfera, etc.), reduciendo el coste de la aplicación directa de la ley de Newton, dependiendo de la geometría del modelo.

La ecuación general 2.91 describe la forma integral de la ley de Gauss y la ecuación 2.92 la forma diferencial. La integral es evaluada sobre una superficie cerrado S , donde $d\mathbf{A}$ es cada una de las porciones infinitesimales de la superficie, \mathbf{n} es el vector normal unitario, perpendicular y que apunta hacia a fuera de la superficie, \mathbf{g} es la aceleración gravitatoria que siempre apunta a la masa, G es la constante de gravitación universal y M_T es la masa encerrada en la superficie S .

$$\oint_S \mathbf{g} \cdot \mathbf{n} d\mathbf{A} = -4 \pi G M_T \quad (2.91)$$

$$\nabla \cdot \mathbf{g} = -4 \pi G \rho \quad (2.92)$$

El coste computacional de este algoritmo esta impuesto por el cálculo de la masa encerrada. Para ello, la práctica habitual es ordenar las partículas en función del radio, una vez ordenadas solo se tiene que recorrer la lista e ir asignando la aceleración gravitatoria según la masa acumulada en el radio de cada partícula. Por tanto, el coste computacional esta delimitado por el algoritmo de ordenación de partículas que, si se escoge un algoritmo eficiente como *Quicksort*, se sitúa en $\theta(n \cdot \log n)$.

La integral 2.91 se puede resolver para cada una de las geometrías simétricas interesantes para los cálculos con SPH, obteniendo los resultados de la tabla 2.7.

Cilindro(2D):	$\mathbf{g}_i = \frac{2 G m(r_i^{CM})}{r_i^{CM}} \hat{\mathbf{r}}_i^{CM} = 2 G m(r_i^{CM}) \frac{\mathbf{r}_i^{CM}}{(r_i^{CM})^2} \quad (2.93)$
Esfera(3D):	$\mathbf{g}_i = \frac{G m(r_i^{CM})}{(r_i^{CM})^2} \hat{\mathbf{r}}_i^{CM} = G m(r_i^{CM}) \frac{\mathbf{r}_i^{CM}}{(r_i^{CM})^3} \quad (2.94)$

Cuadro 2.7: Ecuaciones para la aplicación de la ley de Gauss en superficies simétricas, donde r_i^{CM} es el radio de la partícula i al centro de masas CM y $m(r_i^{CM})$ es la masa encerrada en el radio al centro de masas.

Gravedad aproximada calculada con el potencial gravitatorio

Este método se puede ver en detalle en *García-Senz y colaboradores, 2009* [27], pero aplicado a una versión axisimétrica del SPH. En el código ISFAA utiliza, habitualmente, la implementación de la gravedad con la aproximación a \mathbf{g}_i mediante el potencial gravitatorio, según se describe en las ecuaciones 2.11 y 2.30. Aplicando la discretización SPH obtenemos la ecuación 2.95.

$$\mathbf{g}_i = -\nabla_i \phi_i = \frac{1}{\rho_i} \sum_{j=1}^N m_j (\phi_i - \phi_j) \nabla_i \tilde{\mathbf{W}}_{ij} \quad (2.95)$$

donde el sumatorio se extiende por los vecinos de la partícula y no por el resto de partículas.

En el cálculo del potencial gravitatorio (ecuación 2.89) intervienen todas las partículas del modelo y tiene un coste computacional relativamente alto de $\theta(n^2)$, similar al coste de la gravedad exacta. Utilizar la ecuación 2.95 es muy interesante para un código implícito porque, al incorporar

la gravedad con esta aproximación, la matriz sigue siendo dispersa.

Además calcular el gradiente del potencial de forma aproximada reduce el coste computacional a $\theta(n \cdot \log n)$, en los siguientes apartados se detalla la forma de hacerlo mediante la expansión multipolar. Si en el futuro se encuentran métodos para invertir la matriz a un coste computacional menor que el actual la gravedad no sería el nuevo límite. Es necesario hacer notar que en esta aproximación se pierde precisión en la parte central y en la parte mas exterior del modelo. A pesar de que este procedimiento funciona bien y es computacionalmente eficiente tiene la desventaja de que, por construcción, la ecuación 2.95 no es completamente simétrica.

Expansión multipolar Actualmente esta implementado en el código ISFAA pero no se utiliza porque el coste computacional de la inversión de la matriz es superior al del cálculo de la gravedad, sin embargo en futuras mejoras del código este coste podría bajar y ser de gran utilidad por su impacto en el tiempo de computación. En los códigos astrofísicos de SPH para un número de partículas $n \geq 10^5$, se buscan algoritmos con un coste computacional no mayor de $\theta(n \cdot \log n)$. Para conseguir costes algorítmicos menores y con el fin de preservar la localicidad del SPH, se han desarrollado varias aproximaciones al potencial gravitatorio. Una de las más usadas se basa en la expansión multipolar usando árboles jerárquicos: binarios en 1D, cuadráticos en 2D y octales en 3D (*Hernquist y Katz, 1989* [38]), basándose en el algoritmo de *Barnes y Hut, 1986* [8], con este algoritmo se reduce el tiempo de cálculo de $\theta(n^2)$ a $\theta(n \cdot \log n)$. Comienza mediante la construcción de un cubo que encierra todo el sistema (denominado nodo raíz). De forma recursiva se subdividen las componentes dimensionales en mitades, obteniendo 2^d sub-celdas diferenciadas y agregándolas, en forma de árbol, a la estructura anterior (denominado nodo padre). La división de las sub-celdas continua de forma recursiva si las celdas contienen más de una partícula, y se detiene en el caso de que la sub-celda no contenga partículas o si contiene una única partícula (denominado nodo hoja). Se puede observar un ejemplo en 2D en la figura 2.2.

Imponiendo un criterio entre el tamaño del cluster de partículas y la distancia a la partícula que se evalúa, se puede sustituir la interacción individual por la contribución colectiva de todas las partículas del cluster. De esta forma el potencial gravitatorio de clusters distantes se aproxima por la expansión multipolar. Tomando la representación integral del potencial gravitatorio (ecuación 2.86), suponemos que el movimiento de la partícula evaluada es tal que nunca se encuentra cerca de las fuentes de potencial, de tal forma que, $r' \ll r$. De esta forma, se puede expandir el denominador mediante una serie de Taylor sobre \mathbf{r} (véase sección 2.2.1).

$$\frac{1}{|\mathbf{r}' - \mathbf{r}|} \approx \frac{1}{r} + \sum_i r'_i \frac{\partial(1/r)}{\partial r_i} + \frac{1}{2} \sum_i \sum_j r'_i r'_j \frac{\partial^2(1/r)}{\partial r_i \partial r_j} - \frac{1}{6} \sum_i \sum_j \sum_k r'_i r'_j r'_k \frac{\partial^3(1/r)}{\partial r_i \partial r_j \partial r_k} \quad (2.96)$$

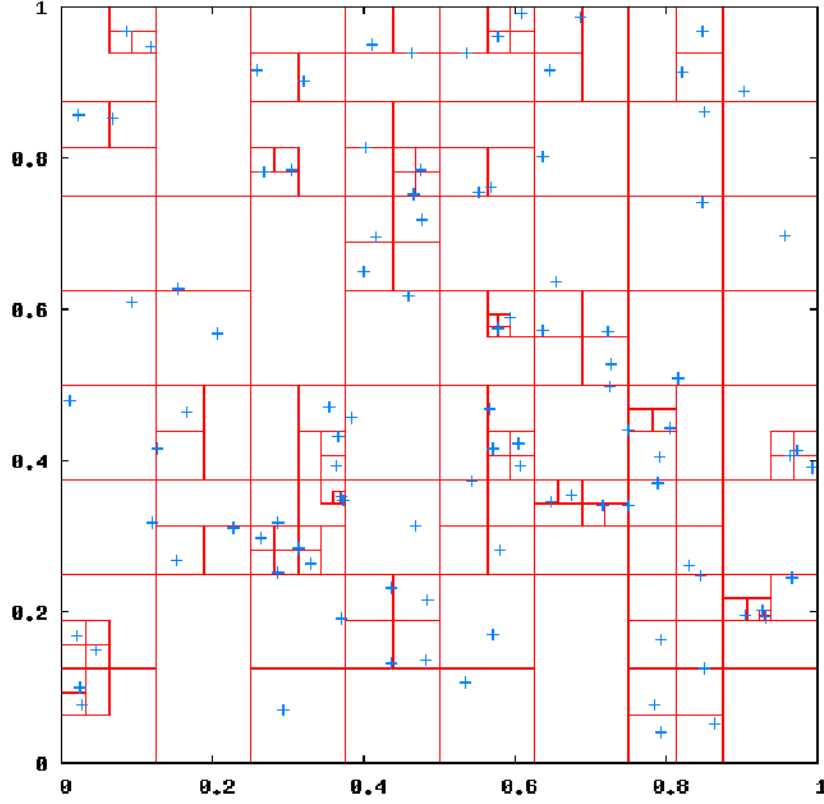


Figura 2.2: División espacial en 2D de 100 partículas colocadas aleatoriamente, mediante el árbol cuadrático del algoritmo de Barnes-Hut.

Aproximación del potencial gravitatorio en clusters (ϕ_S) con la ecuación de *Poisson*

Si denominamos cluster a cada uno de los sub-árboles del algoritmo de *Barnes-Hut*, según se describe en *Cabezón, 2010* [14], suponiendo que la partícula i no forma parte del cluster S y esta lo suficientemente lejos, el potencial gravitatorio de la partícula i contiene la contribución de todos los potenciales de las partículas del cluster S , aproximado por ϕ_S (ecuación 2.97).

$$\phi_S = \sum_{j \in S} \frac{m_j}{|\mathbf{r}_i - \mathbf{r}_j|} \simeq \frac{M_{T_S}}{|\mathbf{r}_i - \mathbf{r}_S^{CM}|} + \frac{1}{2} \frac{Q_{l,m}^S (\mathbf{r}_i - \mathbf{r}_S^{CM})_l (\mathbf{r}_i - \mathbf{r}_S^{CM})_m}{|\mathbf{r}_i - \mathbf{r}_S^{CM}|^5} + \dots \quad (2.97)$$

donde M_{T_S} es la masa del cluster, \mathbf{r}_S^{CM} es el vector radio al centro de masas del cluster, $Q_{l,m}^S$ es el tensor momento cuadrupolar sobre \mathbf{r}_S^{CM} del cluster y l, m son índices con un rango entre 1 y 3 en 3D. Para conseguir una mejor precisión en la aproximación podrían añadirse a la ecuación 2.97 más términos siendo el siguiente el momento octupolar, pero para la mayoría de aplicaciones es suficientemente preciso el valor de ϕ_S hasta el término cuadrupolar. Es necesario observar que el momento dipolar no contribuye en la ecuación, y por tanto es necesario establecer un criterio para el calculo de la gravedad, que elija cuando las partículas se encuentran lejos y su contribución puede ser calculada por la expansión multipolar o por el contrario cuando se

encuentran cerca y se debe calcular la gravedad exacta de forma directa. La representación de esta distancia puede observarse claramente en la figura 2.3.

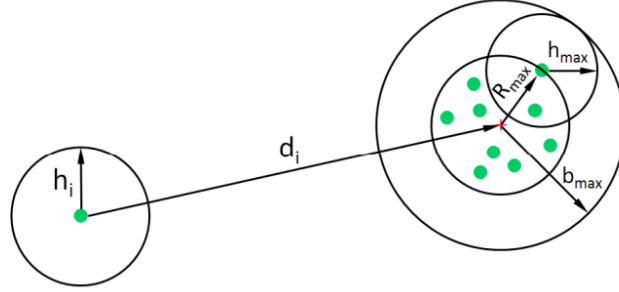


Figura 2.3: Representación esquemática del cluster de partículas S , donde la contribución del cluster al potencial gravitatorio de la partícula i se puede tratar con la expansión multipolar. Donde, d_i es la distancia entre la partícula y el centro de masas del cluster, R_{Max} es el radio desde el centro de masas del cluster a la partícula más alejada del cluster, h_{Max} es la h más grande de todas las partículas del cluster y $b_{Max} = R_{Max} + h_{Max}$ es un parámetro que garantiza que la partícula i y las partículas del cluster no comparten vecinos.

Se puede establecer el criterio de selección de la forma siguiente, donde el parámetro η_{M_p} es un parámetro de tolerancia y usualmente tiene su valor en el rango $[0.5, 0.7]$

$$\left. \begin{array}{ll} b_{Max} + h_i \leq \eta_{M_p} \cdot d_i, & \text{Expansión multipolar} \\ b_{Max} + h_i > \eta_{M_p} \cdot d_i, & \text{Suma directa} \end{array} \right\} \quad (2.98)$$

La ecuación 2.97 representa una gran ventaja ya que su evaluación siempre utiliza la misma cantidad de tiempo independientemente del número de partículas del cluster S . Cuando se construye la estructura de árbol se calcula de forma simultanea los momentos multipolares de cada nivel de forma eficiente, a la vez que se calculan los vecinos de cada partícula.

Dada fuente compacta la ecuación de *Poisson* es formalmente igual a la del potencial gravitatorio Newtoniano, pero con diferentes fuentes compactas. Afortunadamente, la estructura de árbol *Barnes-Hut* puede resolver las ecuaciones de *Poisson* para el potencial gravitatorio a partir de una agrupación eficaz de las partículas y el cálculo de los momento multipolares. Si la partícula que se esta evaluando es el punto de origen se obtiene la ecuación 2.99.

$$\nabla^2 \phi = -4 \pi G \rho \quad (2.99)$$

La aproximación multipolar es lo suficientemente buena aproximando hasta el término cuadrupolar, por tanto, solo necesitamos los tres primeros momentos de los clusters de partículas

(M :Monopolo, D_l :Dipolo y $Q_{l,m}$:Cuadripolo).

$$M^i = \sum_{i=1}^n \frac{m_i}{\rho_i} \rho_i \quad (2.100)$$

$$D_l^i = \sum_{i=1}^n \frac{m_i}{\rho_i} \rho_i x_{i,l} \quad (2.101)$$

$$Q_{l,m}^i = \sum_{i=1}^n \frac{m_i}{\rho_i} \rho_i \left(x_{i,l} x_{i,m} - \frac{1}{3} \delta_{l,m} \mathbf{r}_i^2 \right) \quad (2.102)$$

donde los sumatorios se realizan sobre todas las partículas de los clusters, x_i^l es la componente l del vector de posición de la partícula donde l va de 1 a 3 en 3D y x_i^m es la componente m del vector de posición de la partícula donde m va de 1 a 3 en 3D.

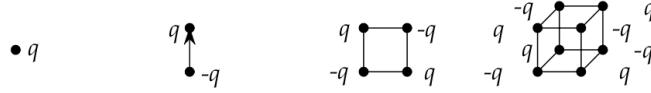


Figura 2.4: Tipos de multipolo: 1) Monopolo, 2) Dipolo, 3) Cuadripolo, 4) Octupolo.

Usando estos momentos se calcula la aproximación al potencial gravitatorio (ecuación 2.103) y sus derivadas espaciales, primera (ecuación 2.104) y segunda (ecuación 2.105), en cada cluster, valores que serán utilizados en la resolución del esquema. En estas ecuaciones, r es la distancia al centro de masas del cluster, por tanto el momento dipolar D_l es automáticamente cero y, por tanto, en las ecuaciones anteriores los momentos dipolares se igualan a cero. Por contra, con el fin de referenciar las posiciones del centro de masas del cluster se tiene que realizar un recorrido inicial adicional del árbol (coste computacional $\theta(n \cdot \log n)$), pero gran parte de los cálculos se distribuyen entre los dos recorridos del árbol y los cálculos adicionales no suponen una sobrecarga importante.

$$\phi = \frac{M}{r} - \frac{D_l x_l}{r^3} + \frac{1}{2} \frac{x_l x_m Q_{l,m}}{r^5} \quad (2.103)$$

$$\frac{\partial \phi}{\partial x_k} = -\frac{M}{r^3} x_k - \frac{D_k}{r^3} + 3 \frac{D_l x_l}{r^5} x_k + \frac{x_m Q_{k,m}}{r^5} - \frac{5}{2} \frac{x_l x_m Q_{l,m}}{r^7} x_k \quad (2.104)$$

$$\begin{aligned} \frac{\partial \phi}{\partial x_p \partial x_k} = & -\frac{M}{r^3} \delta_{p,k} + 3 \frac{M}{r^5} x_p x_k \\ & + \frac{3}{r^5} (D_p x_k + D_k x_p - \delta_{p,k} D_l x_l) + 15 \frac{D_l x_l}{r^7} x_p x_k \\ & + \frac{Q_{p,k}}{r^5} - 10 \frac{x_l x_p Q_{l,k}}{r^7} - \delta_{p,k} \frac{5}{2} \frac{x_l x_m Q_{l,m}}{r^7} + \frac{35}{2} \frac{x_l x_m Q_{l,m}}{r^9} x_p x_k \end{aligned} \quad (2.105)$$

El sistema multipolar se encuentra implementado, dando lugar a un código con más opciones

y más versátil.

Suavizado de la masa En *Evrard, 1988* [25] podemos encontrar un ejemplo de suavizado del potencial gravitatorio que intenta evitar la inestabilidad numérica en el problema de la colisión de N-cuerpos, en este caso con un parámetro ε que evita que se anule el denominador de la ecuación 2.89. Como se ha comentado anteriormente, la aproximación descrita en este documento pierde efectividad en la parte central y en la parte mas externa de los modelos. Siguiendo una filosofía similar, se pude tratar de minimizar este hecho suavizando el potencial gravitatorio minimizando la masa que interviene en el cálculo cuando las partículas son vecinas. La masa modificada m_{ij} de la partícula en interacción m_j es corregida a la baja según el factor η_m (ecuaciones 2.106 y 2.107), donde d es el número de dimensiones.

$$\eta_m = \min \left(1, \left(\frac{r_{ij}}{2 h_i} \right)^d \right) \quad (2.106)$$

$$m_{ij} = \eta_m \cdot m_j \quad (2.107)$$

2.2.5. Viscosidad artificial

En los código hidrodinámicos SPH es necesario incorporar la viscosidad artificial como formalismo para gestionar los choques entre partículas. La viscosidad artificial ayuda a controlar el ruido numérico que se produce en SPH a causa del aumento de entropía que se observa en los choques. En *Von Neumann y Richtmyer, 1950* se dice que el objetivo de la viscosidad artificial es introducir términos disipativos, de forma artificial, en las ecuaciones para que los choques tengan un frente de onda con espesor comparable con la distancia de resolución de las celdas simuladas; entonces las ecuaciones pueden ser usadas como si no se estuviera produciendo el choque. Básicamente convierte la energía cinética en calor. Aunque todavía está en uso, se ha sustituido progresivamente por esquemas mas eficientes (y sofisticados) basados en resolvedores de Riemann, especialmente en los cálculos multidimensionales utilizando en los códigos eulorianos. Sin embargo, debido a las similitudes entre SPH y la dinámica molecular, la viscosidad artificial sigue siendo usada en la actualidad en la mayoría de las aplicaciones donde interviene la viscosidad física entre los átomos y las moléculas. Aunque existe formulaciones de la viscosidad artificial desarrollada al estilo de los resolvedores de Riemann (*Monaghan, 1997* [60]; *Inutsuka, 2002* [42]; *Cha y Whitworth, 2003* [19]) se ha implementado la formulación más estándar de la viscosidad artifical, tal como aparece en *Monaghan, 1992* [58], ya que es la que se usa habitualmente. Sin embargo, en la literatura podemos consultar multitud de trabajos realizados en el campo durante estos últimos años tratando de mejorar esta formulación (*Hernquist y Katz, 1989* [38]; *Wadsley y colaboradores* [101]; *Springel, 2005* [93]; *Rosswog y Price, 2007* [80]).

Veamos ahora en detalle la formulación estándar de la viscosidad. El término de la viscosidad artificial Q es sumado a la verdadera presión P en la ecuación del momento. La presión “viscosa” es una función de la divergencia de la velocidad que toma un valor alto durante las compresiones. Para simular mejor los choques fuertes y el suavizado de las oscilaciones del fluido en las zonas post-choque se suele definir, adicionalmente, un término cuadrático Q_q proporcional al cuadrado de la divergencia de la velocidad.

Por tanto, la viscosidad artificial viene definida por:

$$Q = Q_{bulk} + Q_q \quad (2.108)$$

$$Q_{bulk} = -\alpha \rho l^2 C_s \nabla \mathbf{v} \quad (2.109)$$

$$Q_q = \beta \rho l^2 (\nabla \mathbf{v})^2 \quad (2.110)$$

donde α, β son parámetros ajustables; C_s es la velocidad del sonido local; l es la longitud de escala sobre la que el choque se propaga ($l \simeq h$)

El valor del gradiente es obtenido siguiendo la ecuación 2.11:

$$\nabla \mathbf{v}_i = -\frac{1}{\rho_i} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \quad (2.111)$$

Nosotros usaremos la variante de la viscosidad artificial descrita en *Monaghan, 1992* [58], que se implementa del siguiente modo:

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{C}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}}, & (\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0) \\ 0, & (\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} \geq 0) \end{cases} \quad (2.112)$$

Donde $\alpha = 1$; $\beta = 2$; $\mu_{ij} = h_{ij} \cdot \frac{(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})}{\mathbf{r}_{ij}^2 + \eta_\mu^2}$; $\eta_\mu = 0.1 \cdot h$

Correcciones de Balsara Para limitar la transferencia espúrea de momento angular que surge en el flujo de cizalla ó “shear flow”, debido a la viscosidad artificial, se utilizan habitualmente las corrección de *Balsara, 1995* [7]; *Steinmetz, 1996* [97]. El objetivo de este factor es eliminar el efecto de la viscosidad artificial cuando hay flujo de cizalla, y está pensado para que actúe solamente cuando se produce compresión. Para ello se introduce un factor f , que se define con una estimación de la cantidad relativa de flujo de cizalla que se produce por el rotacional y la divergencia de la velocidad de las partículas (ecuación 2.113), donde el tercer término del denominador previene que la fracción sea divergente.

$$f_i = \frac{|\nabla \cdot \mathbf{v}_i|}{|\nabla \cdot \mathbf{v}_i| + |\nabla \times \mathbf{v}_i| + \left(0.0001 \frac{C_{s_i}}{h_i}\right)} \quad (2.113)$$

Donde la aproximación SPH de la divergencia y el rotacional (según se define en las ecuaciones 2.12 y 2.13) es:

$$\nabla_i \cdot \mathbf{v}_i = -\frac{1}{\rho_i} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \quad (2.114)$$

$$\nabla_i \times \mathbf{v}_i = \frac{1}{\rho_i} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \times \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \quad (2.115)$$

El valor medio para la interacción de dos partículas contiene el valor anterior de Π_{ij} , donde la ecuación 2.112 se sustituye en las ecuaciones del momento y la energía por el nuevo valor de la viscosidad (ecuación 2.116).

$$\Pi'_{ij} = \Pi_{ij} \frac{(f_i + f_j)}{2} \quad (2.116)$$

2.2.6. Conducción térmica

Es necesario incluir una aproximación del término de conducción térmica en la ecuación de la energía, para resolver los problemas derivados de los flujos de calor emitidos o absorbidos en determinados modelos (absorción o emisión en procesos de excitación o desexcitación de los núcleos, reacciones nucleares, etc.). En la literatura podemos encontrar ejemplos de conducción aplicada a la astrofísica, por ejemplo en simulaciones cosmológicas de formación de clusters (*Jubelgas y colaboradores, 2004* [44]).

Además este término permite suavizar el exceso de calor introducido por la viscosidad artificial en el conocido problema “The wall heating shock” (*Noh, 1987* [67]; Véase sección 4.2). Para obtener esta contribución a la energía es necesario resolver la ecuación siguiente:

$$\left(\frac{du}{dt} \right)_{cond} = \frac{1}{\rho} \nabla \cdot (\kappa \nabla T) \quad (2.117)$$

donde κ es el coeficiente de conductividad térmica.

Para el valor de κ , *Monaghan, 1992* [58] propone usar la ecuación 2.118:

$$\kappa_i = \bar{\rho}_{ij} \bar{C}_{vij} (\bar{h}_{ij} \bar{c}_{ij} + 4 |\mu_{ij}|) \quad (2.118)$$

orientada a eliminar el “Wall-shock problem”.

Donde, \bar{C}_v es el calor específico a volumen constante simetrizado, \bar{c} es la velocidad del sonido local simetrizada y μ_{ij} es un parámetro que proviene de la viscosidad artificial (vease Sec. 2.2.5). Por tanto, podemos observar que el parámetro κ es simétrico y por tanto $\kappa_i = \kappa_j$.

Se puede desarrollar la ecuación 2.117 como sigue:

$$\frac{1}{\rho} \nabla \cdot (\kappa \nabla T) = \frac{1}{2\rho} [\nabla^2 (\kappa T) - T \nabla^2 \kappa + \kappa \nabla^2 T] \quad (2.119)$$

Ahora se utiliza la expresión integral de la derivada del núcleo de interpolación.

$$(\nabla^2 Y)_i = \nabla \cdot (\nabla Y)_i = 2 \sum_{j=1}^N \frac{m_j}{\rho_j} \frac{Y_j - Y_i}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \quad (2.120)$$

Y se aplica la ecuación 2.120, sobre los términos de la parte derecha de la ecuación 2.119:

$$\nabla^2 (\kappa_i T_i) = 2 \sum_{j=1}^N \frac{m_j}{\rho_j} \frac{\kappa_j T_j - \kappa_i T_i}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \quad (2.121)$$

$$T_i \nabla^2 (\kappa_i) = 2 \sum_{j=1}^N \frac{m_j}{\rho_j} \frac{\kappa_j T_i - \kappa_i T_i}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \quad (2.122)$$

$$\kappa_i \nabla^2 (T_i) = 2 \sum_{j=1}^N \frac{m_j}{\rho_j} \frac{\kappa_i T_j - \kappa_i T_i}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \quad (2.123)$$

Después se sustituye en los términos en la ecuación 2.119 y aplicamos la igualdad siguiente:

$$\kappa_i T_i + \kappa_i T_j + \kappa_j T_i + \kappa_j T_j = (\kappa_i + \kappa_j) (T_i - T_j) \quad (2.124)$$

Y por último se obtiene:

$$\left(\frac{du_i}{dt} \right)_{cond} = \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} \frac{(\kappa_i + \kappa_j) (T_j - T_i)}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i \bar{W}_{ij} \quad (2.125)$$

Que coincide con la expresión dada por *Springel, 2012* [95], pero simetrizando el kernel (\bar{W}_{ij}). Si en el esquema de resolución se incluye la temperatura la ecuación 2.125 se modifica según la relación:

$$\left(\frac{dT}{dt} \right) = \frac{1}{C_v} \left(\frac{du}{dt} \right) \quad (2.126)$$

2.2.7. Ecuación de estado (EOS)

La ecuación de estado, o EOS, es una ecuación constitutiva para sistemas físicos que establece una relación entre la presión, la densidad, la energía interna y la composición química. Varios tipos de EOS han sido implementadas en el código ISFAA: 1) EOS proveniente del gas ideal; 2) EOS politrópica; 3) EOS realista adaptada a los interiores estelares. En el apéndice D tenemos las derivadas necesarias para la implementación de las diferentes ecuaciones en el código.

Los modelos de test presentados en este documento utilizan la EOS ideal.

EOS ideal La ecuación de gas ideal aproxima la materia suponiendo que las moléculas del gas son puntuales (es decir, no ocupan un volumen) y considera que las fuerzas entre moléculas son despreciable. Por tanto la ley de gases ideales, en este caso gas perfecto, puede describirse por:

$$PV = nRT \quad (2.127)$$

donde P es la presión, V es el volumen, n es el numero de moles, R es la constante universal de los gases ideales que vale $R = 8.314472 \cdot 10^7$ en $\text{erg/mol} \cdot K$ y T es la temperatura. Además puede expresarse según la ecuación 2.128 general de gas ideal, que es la que se utiliza en el código.

$$P = \rho(\gamma - 1)U \quad (2.128)$$

donde ρ es la densidad, U es la energía interna, γ es el coeficiente adiabático, siendo $\gamma = C_v/C_p$ con un valor $\gamma = 5/3$ para un gas de hidrógeno monoatómico, C_v es la capacidad calorífica a volumen constante, C_p la capacidad calorífica a presión constante y la relación entre ellas es $C_v = C_p - R$.

Esta ecuación de gas ideal no tiene buen comportamiento cuando el volumen es pequeño o la temperatura es baja, ya que las aproximaciones asumidas son incorrectas. Es por eso que su rango de utilización se debe limitar a altas temperaturas y bajas presiones (el volumen es muy grande comparado con el volumen de las moléculas del gas).

EOS politrópica En esta ecuación se establece una relación simple entre la presión y la densidad que es aproximadamente seguida por la materia en condiciones degeneradas, como en el interior de las estrellas masivas (enanas blancas, estrellas de neutrones, etc.).

$$P = K \rho^\Gamma \quad (2.129)$$

donde K es una constante y Γ es la denominada gamma adiabática (que esta relacionado con el índice politrópico $n = 1/(\Gamma - 1)$).

Esta EOS es facil de implementar y tiene buen comportamiento, por eso ha sido ampliamente utilizada en la literatura para llevar a cabo estudios paramétricos en estructuras estelares muy densas. Los resultados, en sistemas no relativistas, son muy aproximados a los obtenidos con EOS realistas. En *Cabezón y García-Senz, 2007* [15] podemos encontrar métodos para hacer la correspondencia entre la EOS politrópica y la realista. Es demasiado simple para reproducir con exactitud los resultados de la EOS realista, pero puede ser usada para estudios preliminares.

EOS realista para objetos compactos En el caso del código ISFAA se ha escogido una ecuación que consiste en una mezcla de iones tratada como un gas ideal con correcciones coulombianas, electrones parcialmente degenerados y radiación (*Blinnikov y colaboradores, 2011* [12]). La implementación en detalle puede consultarse en el apéndice D.

2.2.8. Contornos

A pesar de que las superficies libres en SPH no requieren ningún tratamiento especial, si implementamos modelos cerrados, los contornos si que requieren ese tratamiento específico, puesto que SPH trabaja únicamente con interacciones entre partículas (*Rey, 2010* [76]).

Los contornos son todavía un problema abierto en SPH y dependiendo del problema pueden ser difíciles de implementar, pudiendo producirse la interpenetración en el contorno de las partículas pertenecientes al fluido. En *Monaghan, 1994* [59] y *Monaghan, 2005* [61], se deduce la relación de fuerzas que se ejercen las partículas del contorno con las partículas del fluido utilizando la ecuación *Lennard-Jones* aplicada al SPH. Asegurar esta relación es necesario para garantizar el momento lineal y angular del sistema.

Existen muchos métodos disponibles para la implementación de contornos, pero todos tienen sus ventajas y sus inconvenientes, los más habituales son los siguientes:

- Contornos fijos: Se colocan una serie de partículas que permanecen fijas en su posición e interaccionan con el fluido que se acerca.
- Contornos reflexivos: Cuando una partícula se acerca al contorno recibe una fuerza repulsiva de ella misma y las partículas de su entorno pero como si se reflejaran en un espejo, de tal modo que a mayor acercamiento al contorno mayor fuerza repulsiva recibe.
- Contornos periódicos: Se supone que el recinto es invariante frente a traslaciones, de modo que las partículas situadas en un extremo del recinto interaccionan con las partículas del otro lado del recinto.

En sus diferentes versiones, el código ha implementado los tres tipos de contornos aquí descritos.

Contornos fijos Para implementarlo describimos una serie de partículas que forman parte del contorno y actúan sobre las partículas del fluido. Existen dos formas de contornos fijos.

- La más simplificada supone que las partículas se comportan como partículas normales del fluido excepto que no se actualiza su posición ni su velocidad, es decir lo único que variarían es la ecuación de la densidad (o la ecuación de la continuidad) y la presión (la ecuación de estado). Para simular la evolución de un fluido delimitado por unos contornos fijos cerrados, se impone una condición de velocidad específica para todas las partículas que definen el contorno, siendo esta nula o siguiendo algún tipo de movimiento (como un movimiento armónico si queremos que los contornos provoquen fluctuaciones en el fluido, como por ejemplo en pistones).

- Una forma mejorada de contornos fijos es introducir una función en las partículas del contorno, que dependa de la distancia y , que ejerza una fuerza sobre las partículas de su entorno. Es decir, si la partícula i esta a una distancia r de una partícula del contorno, la partícula del contorno ejerce una fuerza total sobre i tal que:

$$F_i = 0,01 C_s^2 \frac{f(r)}{r} \quad (2.130)$$

donde $f(r)$ es una función similar al gradiente del kernel, pero constante cuando la partícula se acerca al contorno y r tiende a 0, para evitar singularidades y para que la partícula no escape del dominio. Hay que hacer notar que el gradiente del kernel se comporta de la misma forma, tiende a 0 cuando la distancia tiende a 0 (figura 2.5).

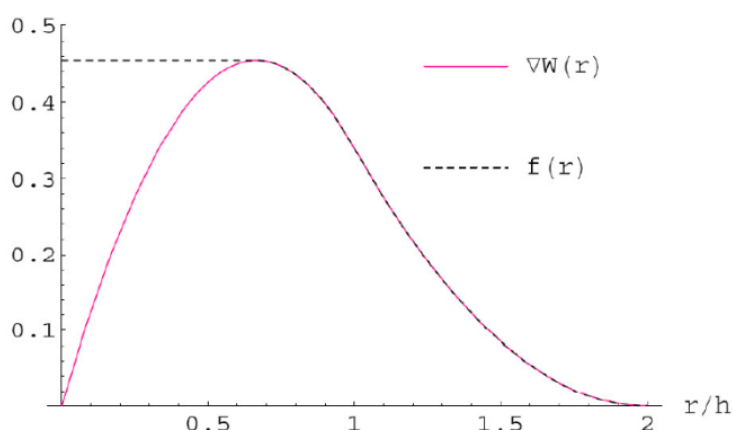


Figura 2.5: Valor de la función del contorno $f(r)$ vs el valor del gradiente del kernel cubico spline $\nabla W(r)$

Contornos reflexivos o periódicos Para implementar este tipo de contornos, a cada iteración del modelo, se deben crear unas partículas “fantasma”.

En el caso reflexivo, los fantasmas reflejan las partículas del fluido cercanas al contorno, de modo que estas partículas tienen la misma densidad y presión, pero la componente normal de la velocidad tiene sentido contrario al de la partícula real. El objetivo es generar una fuerza repulsiva que contenga a las partículas del fluido en el lado correcto del contorno.

En el caso periódico, los fantasmas que ve una partícula cercana al contorno reproducen las propiedades de las partículas del otro lado del contorno como si fuera un fluido continuo o circular. Cuando la posición de una partícula es modificada, si esta se interna en uno de los contornos, se modifica su posición reintroduciendola en el modelo por el lado contrario al contorno invadido. Para ello se le suma o resta a su posición la distancia entre los contornos, en función del lado por donde ha salido. En este caso las partículas mantienen su velocidad.

Estabilidad de los contornos La inclusión de contornos suele constituir nuevas fuentes de inestabilidades, generando aglomeraciones y vacíos de partículas que provocan comportamientos poco realistas. Desde el punto de vista del análisis armónico, las condiciones de contorno pueden generar oscilaciones que decaen al alejarse del mismo. Estas condiciones extra deben ser compatibles con el resto de la formulación o, al menos, producir una influencia débil sobre la solución numérica, en caso contrario, pueden provocar nuevas inestabilidades.

2.2.9. Resumen de las ecuaciones que se utilizarán en ISFAA

El cuadro 2.8 contiene las ecuaciones variacionales, con los términos relativos a la física adicional, que han sido implementadas en el código ISFAA.

$$\mathbf{v} = \frac{d \mathbf{r}_i}{d t} = \frac{\mathbf{r}_i - \mathbf{r}_i^{n-1}}{\Delta t} \quad (2.131)$$

$$\mathbf{a} = \frac{d \mathbf{v}_i}{d t} = - \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \nabla_i \mathbf{W}_{ij}(h_i) + \frac{P_j}{\Omega_j \rho_j^2} \nabla_i \mathbf{W}_{ij}(h_j) \right) + \mathbf{a}_{visco} + \mathbf{a}_{grav} \quad (2.132)$$

$$\dot{U} = \frac{d U_i}{d t} = \frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{U}_{visco} + \dot{U}_{cond} \quad (2.133)$$

$$\dot{T} = \frac{d T_i}{d t} = \frac{1}{\Omega_i \rho_i^2} \cdot \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \cdot \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{T}_{visco} + \dot{T}_{cond} \quad (2.134)$$

$$\dot{\rho} = \frac{d \rho_i}{d t} = \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) \quad (2.135)$$

$$\rho = \sum_{j=1}^N m_j W_{ij}(h_i) \quad (2.136)$$

$$h = \left(\frac{K}{\rho} \right)^{1/d} \quad (2.137)$$

Cuadro 2.8: Discretización de las ecuaciones del SPH. Las ecuaciones son la recopilación de los cuadros 2.4 y 2.6 en su versión variacional agregándoles la física adicional que se utilizará en el código. Donde \mathbf{a}_{grav} es la contribución de la gravedad, que se encuentra definida en el apéndice C.2; $\mathbf{a}_{visco}, \dot{U}_{visco}, \dot{T}_{visco}$ es la contribución a cada ecuación de la viscosidad artificial, que se encuentra definida en el apéndice C.1; $\dot{U}_{cond}, \dot{T}_{cond}$ es la contribución de la conductividad térmica, que se encuentra definida en el apéndice C.3. Las ecuaciones 2.135 y 2.136, son dos versiones para el cálculo de la densidad y, en función del problema a tratar, se escoge cual se utiliza.

2.3. Esquemas

Las ecuaciones básicas de la dinámica de fluidos son utilizadas para determinar el comportamiento de los sistemas físicos y están basadas en las leyes de conservación. La descripción utilizada en SPH es una descripción lagrangiana del sistema donde, como se ha explicado con anterioridad, un conjunto discreto de puntos de interpolación se mueven con el fluido (denominados partículas por tener masa asociada). A cada paso de tiempo las propiedades del fluido se reconstruyen desde este conjunto de puntos.

2.3.1. Tipos de métodos: Directos e iterativos

Cuando se discretiza un sistema de ecuaciones diferenciales habitualmente conduce a un sistema de ecuaciones algebraicas, eventualmente no lineales, fuertemente acoplado. Si suponemos que el sistema es estable analíticamente, la estabilidad numérica está asociada al algoritmo utilizado para resolver el sistema de ecuaciones algebraicas.

Cuando el sistema es lineal puede resolverse, en principio, por métodos directos, otro enfoque es recurrir a los métodos iterativos, cuya estabilidad se manifiesta a través de la convergencia del proceso iterativo.

Cuando el sistema de ecuaciones es no lineal se debe recurrir, necesariamente, a técnicas iterativas. Estas consisten, en general, en plantear y resolver en cada iteración un nuevo sistema lineal. La técnica más difundida para resolución de sistemas no lineales y la que se utiliza en el código SPH ISFAA es el método de *Newton-Raphson*.

Un método iterativo es un método que progresivamente va calculando soluciones aproximadas de un determinado problema. Para ello, repite un mismo proceso de mejora sobre una solución aproximada de tal forma que el resultado obtenido a cada iteración sea una solución más aproximada a la solución analítica. A diferencia de los métodos directos, en los cuales se debe terminar el proceso para tener la respuesta, en los métodos iterativos se puede suspender el proceso al término de una iteración, si la solución aproximada satisface ciertos requisitos predefinidos. Consta de los siguientes pasos: 1) Se inicia la resolución con una solución aproximada o semilla; 2) Se resuelve el sistema para obtener una mejor aproximación partiendo de la aproximación inicial; 3) Se repite el paso anterior pero usando como semilla la aproximación obtenida hasta que la aproximación se acepte como resultado correcto.

Los métodos iterativos se suele usar cuando no se conoce un método directo para obtener la solución de forma exacta o el método directo tiene un coste computacionalmente prohibitivo. Es difícil estimar el coste computacional de un método iterativo, pues no se conoce cuantas

iteraciones se requerirán para obtener una aproximación aceptada. Habitualmente se recurre a calcular el coste computacional por iteración (*iter*) y se añade un parámetro de iteraciones (*i*), de tal forma que el coste total sería $\theta(i \cdot \text{iter})$. Al contrario de lo que pueda parecer, en algunas ocasiones los métodos aproximados permiten obtener un grado de exactitud superior al que se puede obtener empleando los denominados métodos directos. Esto es debido fundamentalmente a los errores de truncamiento que se producen en el proceso.

Ejemplo de resolución: Sistema de una ecuación Si describimos la función $f(x)$:

$$f(x) = x^2 - x - 2 = 0 \quad (2.138)$$

El método directo sería aplicar la fórmula general de las ecuaciones de segundo grado.

$$x = \frac{1 \pm \sqrt{1+8}}{2} = -1, 2 \quad (2.139)$$

El método iterativo *Newton-Raphson*, usado en ISFAA, encuentra la solución usando el punto aproximado menos el cociente entre la función y su derivada en dicho punto.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^2 - x_i - 2}{2x_i - 1} \quad (2.140)$$

Ahora tomamos una aproximación al azar, por ejemplo, $x_0 = 3$ (para $i = 0$), y reintroduciendo la solución en la ecuación de forma iterativa obtenemos lo siguiente.

$$x_1 = x_0 - \frac{x_0^2 - x_0 - 2}{2x_0 - 1} = 3 - \frac{3^2 - 3 - 2}{2 \cdot 3 - 1} \approx 2.2 \quad (2.141)$$

$$x_2 = x_1 - \frac{x_1^2 - x_1 - 2}{2x_1 - 1} = 2.2 - \frac{2.2^2 - 2.2 - 2}{2 \cdot 2.2 - 1} \approx 2.011 \quad (2.142)$$

$$x_3 = x_2 - \frac{x_2^2 - x_2 - 2}{2x_2 - 1} = 2.011 - \frac{2.011^2 - 2.011 - 2}{2 \cdot 2.011 - 1} \approx 2.00004 \quad (2.143)$$

Ejemplo de resolución: Sistema de varias ecuaciones Si tenemos un sistema de n ecuaciones lineales con n incógnitas x_i , donde a_{ij} son los coeficientes de las ecuaciones e incógnitas y b_i los términos independientes de las ecuaciones.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n &= b_3 \\ \cdots &= \cdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n \end{cases} \quad (2.144)$$

El sistema se puede describir utilizando la representación matricial.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \cdots \\ b_n \end{pmatrix} \quad (2.145)$$

Podemos asignar a la matriz de coeficientes el parámetro A , al vector de incógnitas que queremos obtener x , y al vector de términos independientes b , de tal forma que:

$$A \cdot x = b \quad \implies \quad x = A^{-1} \cdot b \quad (2.146)$$

donde la solución implica la inversión de una matriz.

Según se describe en la literatura (*De La Fuente, 1997* [24], *Press y colaboradores, 2007* [71]), etc.), el sistema se puede resolver mediante métodos directos o finitos, como la eliminación de Gauss o la descomposición LU, que permite obtener la solución del sistema de manera directa, o se pueden utilizar métodos iterativos o infinitos que calculan las soluciones del sistema por aproximaciones sucesivas.

2.3.2. Tipos de esquemas: Explícitos e implícitos

La descripción del esquema de resolución que se necesita resolver puede hacerse de forma explícita, implícita o semi-implícita (*Morton y Mayers, 2005* [65]).

Esquema explícito Los esquemas de tipo explícito son más apropiados para modelos que varían rápidamente en el espacio y/o el tiempo y se definen como aquellos en los que el cálculo se realiza en función solamente de los valores que toman las variables en el instante anterior. Es decir, las variables de cada partícula son calculadas, en cada paso de tiempo, en función de sus valores en el instante anterior y de las fuerzas de interacción con sus partículas vecinas. Tienen menor coste computacional que los métodos implícitos, pero es necesario establecer límites en el paso de tiempo para evitar la inestabilidad numérica (Criterio de Courant). Hay que señalar que tienen el problema de no ser incondicionalmente convergentes, es decir la aproximación numérica calculada sin limitar el paso de tiempo diverge exponencialmente de la solución exacta.

Esquema implícito Los esquemas implícitos evalúan los valores de las variables en un instante dado y en una posición concreta, a partir de los valores en las partículas vecinas en el instante

anterior y en el mismo instante. Por tanto, la resolución de las variables involucradas implica la resolución de un sistema de ecuaciones asociado. Esquemas de este tipo son característicos de los métodos de diferencias finitas donde, una vez dividido el dominio en subdominios o partículas en el caso de SPH, las variables de los subdominios son evaluadas a partir de la resolución del sistema de ecuaciones. Si el método implícito llega al mismo resultado que el método explícito, siendo este resultado una aproximación aceptada de la solución exacta, el método es consistente. Si es estable para toda entrada del dominio, es incondicionalmente estable. De ello deducimos que un método implícito, que cumpla los criterios de estabilidad incondicional y consistencia, es incondicionalmente convergente.

Esquema semi-implícito Los esquemas semi-implícitos se obtienen como una combinación entre los métodos explícitos e implícitos. Para ello se introduce un parámetro de ponderación θ con un rango de valores entre $[0, 1]$, que determina el grado de preponderancia entre los dos tipos de esquema, siendo $\theta = 0$ un esquema explícito, $\theta = 1$ un esquema implícito y $\theta = 0.5$ un esquema semi-implícito. El esquema semi-implícito se define en 2.147 y su dominio de soluciones se encuentra descrito en la figura 2.6.

$$\frac{\nu_j^{i+1} - \nu_j^i}{\Delta t} = (1 - \theta) \frac{\nu_{j-1}^i - 2\nu_j^i + \nu_{j+1}^i}{\Delta x^2} + \theta \frac{\nu_{j-1}^{i+1} - 2\nu_j^{i+1} + \nu_{j+1}^{i+1}}{\Delta x^2} \quad (2.147)$$

donde ν es una variable cualquiera, $\Delta t, \Delta x$ son la variación en tiempo y espacio del dominio. Un caso especialmente bueno de la ecuación es cuando $\theta = 0.5$ (esquema semi-implícito), ya que el orden de error de la derivada temporal es $O(\Delta t^2)$, en lugar de $O(\Delta t)$. y mejorando así la precisión de la aproximación. En este caso, tenemos segundo orden de precisión en el tiempo y segundo orden de precisión en el espacio.

2.3.3. SPH explícito: Integración centrada mediante *predictor-corrector*

Los esquemas explícitos son aquellos en los que las variables en un cierto paso de tiempo (Δt^{n+1}) solo dependen de las variables en el tiempo inmediatamente anterior (Δt^n). Es decir, en la descripción explícita, el movimiento de las partículas (\mathbf{r}) queda determinado cuando se conoce la posición inicial (\mathbf{x}_o) y la variación de dicha posición en el tiempo $\left(\frac{\partial \mathbf{x}_o}{\partial t}\right)$, es decir la velocidad (\mathbf{v}). Así mismo, la velocidad queda determinada cuando se conoce la velocidad actual (\mathbf{v}_o) y la variación de velocidad en el tiempo $\left(\frac{\partial v_o}{\partial t}\right)$, es decir la aceleración (\mathbf{a}). Las ecuaciones siguientes muestran las variaciones en posición, velocidad y densidad del esquema explícito a

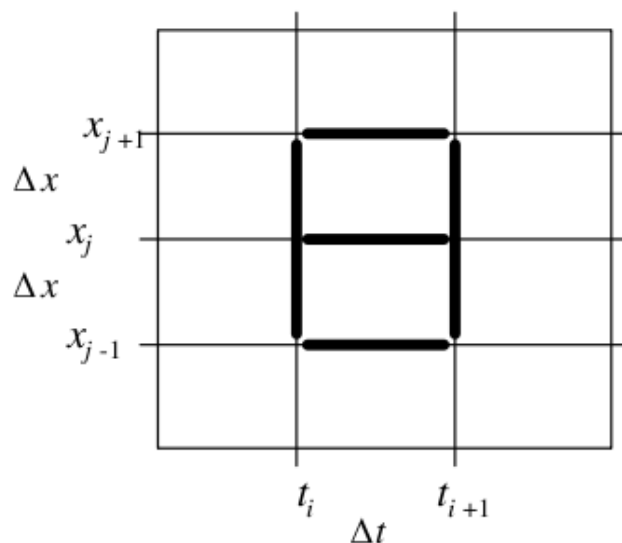


Figura 2.6: Discretización del dominio en el espacio y en el tiempo de un modelo en 1D.

cada paso de tiempo:

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \mathbf{v}^n \Delta t + \frac{1}{2} \mathbf{a}^n \Delta t^2 \quad (2.148)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \mathbf{a}^n \Delta t \quad (2.149)$$

$$\rho^{n+1} = \rho^n + \frac{\partial \rho^n}{\partial t} \Delta t \quad (2.150)$$

Los esquemas explícitos son más apropiados para simular modelos rápidamente variables en el tiempo y por eso en astrofísica se utiliza frecuentemente para reproducir choques o procesos explosivos. El método es sencillo de por sí, pero tiene un problema relacionado con el paso de tiempo. Para mantener la estabilidad su paso de tiempo se encuentra limitado por la condición de Courant.

Condición de Courant en SPH En los esquemas explícitos el paso de tiempo que da lugar a la convergencia está limitado por la distancia (Δx) que puede recorrer una onda sónica cuando atraviesa una celda de resolución del modelo (ecuación E.1). Cuando la distancia entre celdas se reduce, el límite para el intervalo de tiempo también se reduce (figura E.1). Se debe escoger el paso de tiempo de forma que sea lo suficientemente pequeño para conservar la estabilidad, es decir, que para SPH se ha de cumplir la condición de Courant de la forma siguiente.

$$\Delta t \leq \min \left(\frac{h_i}{|C_{s_i}|} \right)_{i=1}^n \quad (2.151)$$

donde el dominio de dependencia de una variable en un esquema explícito está restringido por la velocidad de propagación de la información $|C_s|$ la cual define el dominio de dependencia de la solución exacta, siendo C_s la velocidad del sonido local de cada partícula. En SPH el tamaño de la celda viene determinada por la longitud de suavizado de cada partícula, con una resolución radial habitual por celda de $2h$.

En la literatura se establece una ecuación más ajustada para la condición de Courant (ecuación 2.152), suponiendo $v_i \ll C_{s_i}$.

$$\Delta t \leq t_{Cou} = \eta_{Cou} \cdot \text{Min} \left(\frac{h_i}{C_{s_i}} \right)_{i=1}^n \quad (2.152)$$

donde t_{Cou} es el tiempo de Courant y η_{Cou} es una constante de seguridad para reducir el tiempo escogido, cuyo valor suele estar en el rango $[0.2, 0.5]$. En ISFAA se escoge un valor de $\eta_{Cou} = 0.4$, cercano al central del intervalo, para comparar con los pasos de tiempo implícito. Se ha tomado este valor para que difiera poco al comparar el resultado de ISFAA con elecciones de otros autores. La condición de Courant restringe de forma drástica el paso de tiempo y sería deseable usar métodos implícitos para evitarla.

Integración temporal en SPH En la versión explícita del SPH, utilizado en el GAA, se utiliza una integración centrada.

Usando la idea de dos pasos de tiempo aplicamos un esquema centrado, con paso de tiempo variable, de tal forma que se utiliza un esquema predictor-corrector parecido al expuesto en *Serna y colaboradores, 1996* [91] que consigue buenas aproximaciones y conserva con mucha precisión la energía y el momento angular del sistema. Para ello, calculamos las variables a un tiempo $t^{n+\Delta t/2}$ para corregir las variables en el tiempo $t^{n+\Delta t}$.

1. Con un paso de tiempo $\Delta t/2$, se calculan las variables a tiempo $t^{n+\Delta t/2}$.

- a) Se crea el árbol de partículas
- b) Se recorre el árbol calculando los vecinos, los momentos multipolares (para aproximar el potencial gravitatorio y calcular la gravedad, véase sección 2.2.4) y la densidad SPH (ecuación 2.55).
- c) Como ahora tenemos la densidad calculamos la ecuación de estado escogida (véase sección 2.2.7) y se obtiene la presión.
- d) Se calcula la ecuación del momento y la energía, que incluye el cálculo de la viscosidad artificial y la conducción térmica.

e) Se actualizan las posiciones, velocidades, energías y longitud de suavizado.

$$\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\partial \mathbf{v}^n}{\partial t} \frac{\Delta t}{2} \quad (2.153)$$

$$\mathbf{r}^{n+1/2} = \mathbf{r}^n + \left(\frac{\mathbf{v}^n + \mathbf{v}^{n+1/2}}{2} \right) \frac{\Delta t}{2} \quad (2.154)$$

$$U^{n+1/2} = U^n + \frac{1}{2} \frac{\partial U^n}{\partial t} \frac{\Delta t}{2} \quad (2.155)$$

$$h^{n+1/2} = h^n + \frac{1}{2} \left(1 + (2^{\eta_h} - 1) \frac{n_{v_\phi}}{n_{v_{fac}}^{n+1/2}} \right)^{1/\eta_h} \quad (2.156)$$

2. Con un paso de tiempo Δt , se calculan las variables a tiempo $t^{n+\Delta t}$.

a) Se crea el árbol de partículas

b) Se recorre el árbol calculando los vecinos, los momentos multipolares (para aproximar el potencial gravitatorio y calcular la gravedad, véase sección 2.2.4) y la densidad SPH (ecuación 2.55).

c) Como ahora tenemos la densidad calculamos la ecuación de estado escogida (véase sección 2.2.7) y se obtiene la presión.

d) Se calcula la ecuación del momento y la energía, que incluye el cálculo de la viscosidad artificial y la conducción térmica.

e) Se actualizan las posiciones, velocidades, energías y longitud de suavizado.

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\partial \mathbf{v}^{n+1/2}}{\partial t} \Delta t \quad (2.157)$$

$$\mathbf{r}^{n+1} = \mathbf{r}^n + \left(\frac{\mathbf{v}^n + \mathbf{v}^{n+1}}{2} \right) \Delta t \quad (2.158)$$

$$U^{n+1} = U^n + \frac{1}{2} \frac{\partial U^{n+1/2}}{\partial t} \Delta t \quad (2.159)$$

$$h^{n+1} = h^n + \frac{1}{2} \left(1 + (2^{\eta_h} - 1) \frac{n_{v_\phi}}{n_{v_{fac}}^{n+1}} \right)^{1/\eta_h} \quad (2.160)$$

3. Se incrementa el tiempo $t^{n+1} = t^n + \Delta t^n$ y establece el nuevo paso de tiempo Δt^{n+1} .

4. Se almacenan los datos, si es necesario, y si no se cumple la condición de salida del programa se vuelve al paso 1.

Es cierto que se podrían implementar otros esquemas de integración más complejos como un Runge-Kutta de cuarto orden, pero como en estos métodos se trabaja con grandes cantidades de partículas y las ecuaciones a integrar son exactas solo hasta segundo orden, no resulta conveniente incrementar la carga computacional del código sin obtener beneficios.

2.3.4. SPH implícito: Integración *Newton-Raphson*

Un método implícito requiere la resolución de un sistema de ecuaciones no trivial a cada paso de tiempo. Cuando se establece un esquema implícito de resolución es necesario analizar que magnitudes evolucionarán con el esquema (variables independientes) y que magnitudes serán recalculadas a cada paso de tiempo (variables dependientes). En función de estas decisiones se seleccionarán las diferentes ecuaciones del sistema.

En este apartado se presentan tres tipos de esquemas implícitos, que dependen de las variables independientes que se escojan y que generan tres sistemas de ecuaciones diferentes. Los esquemas serán denominados: esquema normal, esquema reducido y esquema inmóvil. Si se quiere que la resolución del sistema de ecuaciones sea computacionalmente eficiente es necesario descartar los métodos directos y resolverlo mediante un método iterativo. En Knapp, 2000 [47] se desarrolla un esquema implícito y se analizan diferentes métodos iterativos llegando a la conclusión de que por precisión y simpleza la elección debe ser el método *Newton-Raphson*, y por tanto, después de que nuestro propio análisis coincida con el de Knapp se escogió este método para implementar el código ISFAA. Este método iterativo permite aproximar una función derivable del tipo $f(x) = 0$. Para conseguirlo se parte de una aproximación inicial x_0 y se construye una sucesión de aproximaciones de forma recurrente, tal que:

$$\Delta x + \frac{f(x_n)}{f'(x_n)} = 0 \quad \implies \quad \Delta x = -\frac{f(x_n)}{f'(x_n)} \quad \implies \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.161)$$

Ejemplos de esquemas implícitos

En este apartado mostraremos tres ejemplos de sistemas implícitos: 1) Implícito con una variable; 2) Semi-implícito con una variable; 3) Implícito con varias variables. El objetivo es familiarizarnos con el montaje necesario para los esquemas que se implementarán en el código, siendo este del tipo semi-implícito con varias variables.

Ejemplo de discretización implícito con una variable Si discretizamos la ecuación tomando la aproximación *backward* de la derivada temporal (ecuación 2.31) y la aproximación *centrada* de la segunda derivada espacial (ecuación 2.34). Siguiendo la misma deducción, la solución en diferencias es la siguiente:

$$\frac{\nu_j^{i+1} - \nu_j^i}{\Delta t} = \frac{\nu_{j+1}^{i+1} - 2\nu_j^{i+1} + \nu_{j-1}^{i+1}}{\Delta x^2}, \quad \implies \quad \nu_j^i = -\frac{\Delta t}{\Delta x^2} \nu_{j-1}^{i+1} + \left(1 + 2 \frac{\Delta t}{\Delta x^2}\right) \nu_j^{i+1} - \frac{\Delta t}{\Delta x^2} \nu_{j+1}^{i+1} \quad (2.162)$$

donde $\alpha = \Delta t / \Delta x^2$. Ahora, la solución en un punto no depende exclusivamente de la solución obtenida en el instante anterior, sino que depende también de la solución en otros puntos en el

mismo instante. Igual que en el caso anterior tenemos primer orden de precisión en el tiempo y segundo orden de precisión en el espacio.

La ecuación 2.162 requiere la resolución de un sistema de ecuaciones representado mediante un sistema matricial (ejemplo de resolución de un sistema de ecuaciones 2.144) , que contiene las evaluaciones de la variable ν en todos los puntos de discretización del dominio y dando un único resultado gracias a que la matriz es invertible.

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & \cdots & 0 & 0 \\ 0 & -\alpha & 1+2\alpha & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1+2\alpha & -\alpha \\ 0 & 0 & 0 & \cdots & -\alpha & 1+2\alpha \end{pmatrix} \cdot \begin{pmatrix} \nu_1^{i+1} \\ \nu_2^{i+1} \\ \nu_3^{i+1} \\ \cdots \\ \nu_{n-1}^{i+1} \\ \nu_n^{i+1} \end{pmatrix} = \begin{pmatrix} \nu_1^i \\ \nu_2^i \\ \nu_3^i \\ \cdots \\ \nu_{n-1}^i \\ \nu_n^i \end{pmatrix} + \alpha \begin{pmatrix} \nu_1^{i+1} \\ 0 \\ 0 \\ \cdots \\ 0 \\ \nu_n^{i+1} \end{pmatrix}$$

Se puede observar que las ecuaciones para el cálculo de la solución están acopladas y cualquier cambio en algún punto j afecta a la solución en el resto de puntos en el mismo instante.

Ejemplo de discretización semi-implícito con una variable Según se detalla en *Menéndez, 2010* [56], si tenemos el siguiente ejemplo con una ecuación diferencial de difusión (ecuación 2.163), tal que, la variable ν depende del tiempo y de la posición en 1D $\nu(t, x)$, π es una constante positiva que representa la viscosidad cinemática, donde $t \geq 0$ y $-\infty < x < \infty$

$$\frac{\partial \nu}{\partial t} = \pi \frac{\partial^2 \nu}{\partial x^2} \quad (2.163)$$

Si discretizamos la ecuación diferencial 2.163 de forma *centrada* (véase ecuación 2.147) obtenemos la ecuación 2.164. Si se escoge $\theta = 0$ o 1 el esquema es de primer orden $\varepsilon = O(\Delta t) + O(\Delta x^2)$. Además, si se escoge $\theta = 0.5$ el esquema es de segundo orden $\varepsilon = O(\Delta t^2) + O(\Delta x^2)$.

$$\frac{\nu_j^{i+1} - \nu_j^i}{\Delta t} = \pi \left[(1 - \theta) \frac{\nu_{j-1}^i - 2\nu_j^i + \nu_{j+1}^i}{\Delta x^2} + \theta \frac{\nu_{j-1}^{i+1} - 2\nu_j^{i+1} + \nu_{j+1}^{i+1}}{\Delta x^2} \right] \quad (2.164)$$

Utilizando el método de *von Neumann* se obtiene el siguiente factor de amplificación (*Richtmyer y Morton, 1967* [77]), donde $r = \pi (\Delta x / \Delta t)$.

$$\xi_m = \frac{1 - 2r(1 - \theta)(1 - \cos m\Delta x)}{1 + 2r\theta(1 - \cos m\Delta x)} \quad (2.165)$$

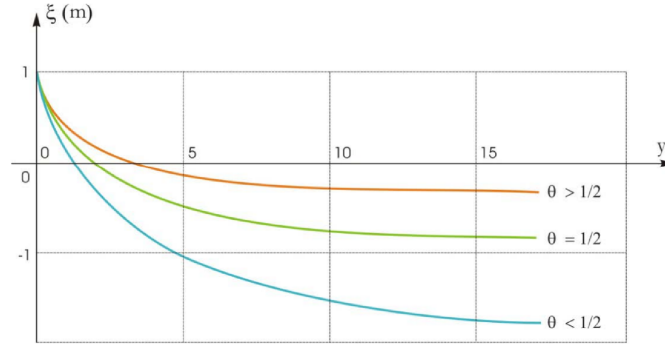


Figura 2.7: Factor de amplificación del esquema 2.164, donde ξ_m es la variación respecto $y = 2r(1 - \cos m\Delta x)$, para distintos valores de θ

En la figura 2.7 se observa que para todo valor de m , si $\theta \geq 0.5$ es esquema es incondicionalmente estable, en caso contrario $0 \leq \theta < 0.5$, siempre decrece por debajo de -1 para valores grandes de y .

Esto determina la restricción designada por la ecuación 2.166 para $\theta > 0$ y $r \leq 1/2$ para $\theta = 0$ (figura 2.8).

$$r \leq 1/(1 - 2\theta) \quad (2.166)$$

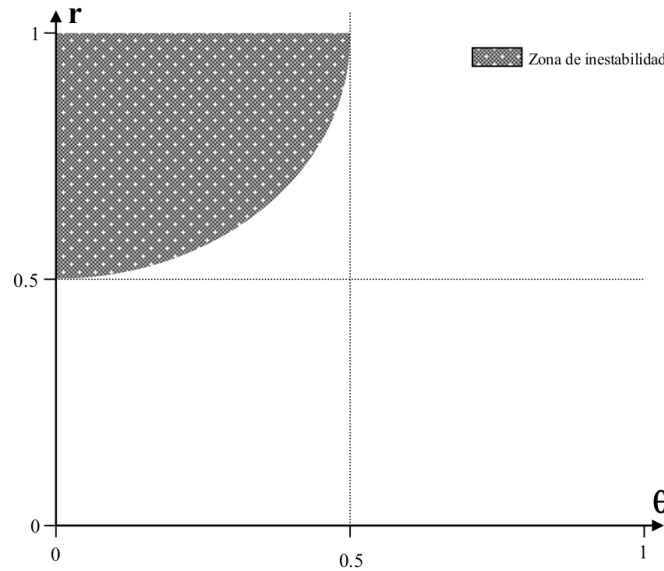


Figura 2.8: Estabilidad numérica del esquema 2.164, donde $r = \pi\Delta x/\Delta x^2$.

Los análisis realizados determinan la estabilidad del problema en diferencias, a ello hay que agregar la estabilidad del algoritmo particular mediante el cual se implementa la resolución del

problema. Esta cuestión es especialmente relevante en el caso de esquemas numéricos implícitos, ya que conducen a sistemas de ecuaciones acopladas. Las condiciones para que estos algoritmos permanezcan estables se describen en *Dahlquist y Bjorck, 2003* [23].

Este ejemplo muestra que se pueden construir esquemas implícitos incondicionalmente estables. A pesar de que no es necesario limitar el valor de Δt , para garantizar la precisión de la solución numérica restricciones como la expuesta en la ecuación 2.166 son necesarias. En el problema de difusión que se está considerando, los efectos difusivos se hacen significativos sobre una distancia Δx en tiempos del orden de $\Delta x^2/\nu$, lo cual significa que r debe de ser del orden de 1 para seguir con precisión la solución analítica. Es decir, que no se puede elegir un paso temporal Δt demasiado distinto al que resultaría para un esquema explícito, lo cual cuestiona la aparente ventaja del método implícito, ya que éste requiere en general muchas más operaciones por paso de tiempo. En este caso, mantener la precisión requiere calcular un paso de tiempo pequeño frente a las escalas activas. En cambio, para mantener la estabilidad de un esquema explícito es necesario restringir el paso, en relación al tiempo de Courant, aunque este tiempo sea mucho menor que el de las escalas activas.

Por tanto, para obtener ventajas de los esquemas implícitos sobre los explícitos requiere que el problema involucre escalas de tiempo activas varios ordenes de magnitud por encima del tiempo de Courant.

Ejemplo de discretización implícito con varias variables El siguiente ejemplo representa un esquema implícito con dos ecuaciones (G_1, G_2) y dos incógnitas (x, y). Mediante el método *Newton-Raphson* se pasa del esquema exacto a otro numérico, donde (G_1^i, G_2^i) son las soluciones del esquema suponiendo que el valor inicial de las variables en la iteración i es (x^i, y^i) , este valor es tomado de forma arbitraria y el esquema se encarga de ajustarlo:

$$\left. \begin{array}{l} G_1(x, y) = 0 \\ G_2(x, y) = 0 \end{array} \right\} \Rightarrow \left. \begin{array}{l} G_1^i(x^i, y^i) = \varepsilon_1^i \quad ; \quad |\varepsilon_1^i| > 0 \\ G_2^i(x^i, y^i) = \varepsilon_2^i \quad ; \quad |\varepsilon_2^i| > 0 \end{array} \right\}$$

El objetivo es que de forma iterativa se obtengan sucesivas correcciones $(\delta x, \delta y)$ para que sumadas a la variables (x, y) permitan cumplir cada vez con más precisión, y por tanto hacer nulas, las ecuaciones (G_1, G_2) . En pocas iteraciones ($i, i+1, i+2, \dots, i+j$) se debe alcanzar una desviación menor que un cierto valor de cota (ε , establecido habitualmente en un rango entre $[10^{-8}, 10^{-4}]$), que acepta la solución como válida, de tal forma que se cumpla el criterio $(|\varepsilon_1^i| \leq \varepsilon_1) \wedge (|\varepsilon_2^i| \leq \varepsilon_2)$.

$$\left. \begin{aligned} G_1^i + \delta G_1^i &= 0 \\ G_2^i + \delta G_2^i &= 0 \end{aligned} \right\} \Rightarrow \left. \begin{aligned} \delta G_1^i &= -G_1^i \\ \delta G_2^i &= -G_2^i \end{aligned} \right\}$$

Por tanto aplicamos diferencias (δ) a las ecuaciones (G_1^i, G_2^i) , sustituimos y montamos un esquema de resolución matricial según el ejemplo 2.144. Este esquema es el estándar que se usa como entrada para las bibliotecas de algoritmos de resolución de sistemas de ecuaciones, como PARDISO o WSMP (vease sección 3.3.2)

$$\left. \begin{aligned} \delta G_1^i &= \left(\frac{\partial G_1^i}{\partial x} \right) \cdot \delta x^i + \left(\frac{\partial G_1^i}{\partial y} \right) \cdot \delta y^i \\ \delta G_2^i &= \left(\frac{\partial G_2^i}{\partial x} \right) \cdot \delta x^i + \left(\frac{\partial G_2^i}{\partial y} \right) \cdot \delta y^i \end{aligned} \right\} \Rightarrow \left. \begin{aligned} \left(\frac{\partial G_1^i}{\partial x} \right) \cdot \delta x^i + \left(\frac{\partial G_1^i}{\partial y} \right) \cdot \delta y^i &= -G_1^i \\ \left(\frac{\partial G_2^i}{\partial x} \right) \cdot \delta x^i + \left(\frac{\partial G_2^i}{\partial y} \right) \cdot \delta y^i &= -G_2^i \end{aligned} \right\}$$

$$\begin{pmatrix} \frac{\partial G_1^i}{\partial x} & \frac{\partial G_1^i}{\partial y} \\ \frac{\partial G_2^i}{\partial x} & \frac{\partial G_2^i}{\partial y} \end{pmatrix} \cdot \begin{pmatrix} \delta x^i \\ \delta y^i \end{pmatrix} = - \begin{pmatrix} G_1^i \\ G_2^i \end{pmatrix}$$

Cuadro 2.9: Ejemplo de esquema de resolución implícito con dos ecuaciones (G_1, G_2) y dos incógnitas (x, y) , en la iteración i .

Después de la resolución del esquema y de obtener $(\delta x^i, \delta y^i)$ se actualizan las variables y se comprueban los nuevos errores $(\varepsilon_1^{i+1}, \varepsilon_2^{i+1})$

$$\left. \begin{aligned} x^{i+1} &= x^i + \delta x^i \\ y^{i+1} &= y^i + \delta y^i \end{aligned} \right\} \Rightarrow \left. \begin{aligned} G_1^{i+1}(x^{i+1}, y^{i+1}) &= \varepsilon_1^{i+1} \quad ; \quad |\varepsilon_1^{i+1}| > 0 \\ G_2^{i+1}(x^{i+1}, y^{i+1}) &= \varepsilon_2^{i+1} \quad ; \quad |\varepsilon_2^{i+1}| > 0 \end{aligned} \right\}$$

- Si los errores cumplen la condición: $(|\varepsilon_1^{i+1}| \leq \varepsilon_1) \wedge (|\varepsilon_2^{i+1}| \leq \varepsilon_2)$, el esquema es correcto y (x^{i+1}, y^{i+1}) , son soluciones aceptadas para el esquema en la iteración $i + 1$.
- Si por el contrario alguno de los nuevos errores no cumple la condición anterior y por lo tanto $(|\varepsilon_1^{i+1}| \geq \varepsilon_1) \vee (|\varepsilon_2^{i+1}| \geq \varepsilon_2)$, no es una solución aceptada y es necesario realizar un nuevo refinamiento $i + 2$ del sistema de ecuaciones con los nuevos valores de las variables (x^{i+1}, y^{i+1}) en $i + 1$.

Es decir, en los sucesivos refinamientos, obtendremos valores $(\delta^i, \delta^{i+1}, \dots, \delta^{i+j})$ que corregirán las variables iniciales (x^i, y^i) hasta que obtengamos valores aceptables para todas las

variables, de forma simultanea, tal que:

$$\left. \begin{aligned} x^{i+j} &= x^i + \delta x^i + \delta x^{i+1} + \dots + \delta x^{i+j} \\ y^{i+j} &= y^i + \delta y^i + \delta y^{i+1} + \dots + \delta y^{i+j} \end{aligned} \right\} \Rightarrow \left. \begin{aligned} G_1^{i+j}(x^{i+j}, y^{i+j}) &\leq \varepsilon_1 \\ G_2^{i+j}(x^{i+j}, y^{i+j}) &\leq \varepsilon_2 \end{aligned} \right\}$$

Esquemas implícitos en SPH

Una vez vistos los ejemplos estamos preparados para introducir las ecuaciones y las variables que se utilizarán en los diferentes esquemas implícitos del código.

Formato de las ecuaciones Si tenemos que ${}^iG_j^n$ es la ecuación j de la partícula i en la iteración n , de forma estándar tendremos que una ecuación cualquiera en formato *Newton-Raphson* evoluciona de la forma siguiente.

$${}^iG_j^{n+1} = {}^iG_j^n + \delta {}^iG_j^n = 0 \quad \Rightarrow \quad \delta {}^iG_j^n = -{}^iG_j^n \quad (2.167)$$

Ecuaciones implícitas centradas en formato *Newton-Raphson* En nuestros sistemas de ecuaciones se tiene hasta nueve ecuaciones ($G_{a_x}, G_{a_y}, G_{a_z}, G_{[U/T]}, G_{v_x}, G_{v_y}, G_{v_z}, G_{[\rho, \rho]}, G_{M_t}$). Si, por ejemplo, se coge la ecuación centrada del momento en el eje x (a_x), en formato *Newton-Raphson* se necesita igualar a cero la ecuación.

$$a_x = \frac{v_x - v_x^{n-1}}{\Delta t} = (1 - \theta) a_x^{n-1} - \theta \left(\frac{\nabla P}{\rho} \right) - \theta g \quad (2.168)$$

$$G_{a_x} = \frac{v_x - v_x^{n-1}}{\Delta t} - (1 - \theta) a_x^{n-1} + \theta \left(\frac{\nabla P}{\rho} \right) + \theta g = 0 \quad (2.169)$$

Desarrollando todas las ecuaciones obtenemos el cuadro 2.10.

Linealización de las ecuaciones Por cada una de las partículas del modelo, las variaciones de cada ecuación dependerán de un número similar de variables independientes, es decir, $G(x, y, z, [U/T], v_x, v_y, v_z, \rho, h)$. Cualquiera de estas ecuaciones puede ser linealizada mediante una expansión de Taylor a primer orden (sección 2.2.1), de donde obtenemos lo siguiente.

$$\delta G = \frac{\partial G}{\partial x} \delta x + \frac{\partial G}{\partial y} \delta y + \frac{\partial G}{\partial z} \delta z + \frac{\partial G}{\partial [U/T]} \delta [U/T] + \frac{\partial G}{\partial v_x} \delta v_x + \frac{\partial G}{\partial v_y} \delta v_y + \frac{\partial G}{\partial v_z} \delta v_z + \frac{\partial G}{\partial \rho} \delta \rho + \frac{\partial G}{\partial h} \delta h = -G \quad (2.181)$$

donde cada una de las ecuaciones iG_j del sistema se ve afectada por la variaciones de cada una de las variables independientes de cada partícula (n_v), es decir, si n es el número partículas $n \cdot n_v$ es el número de variables independientes del sistema $(x^k, y^k, z^k, [U/T]^k, v_x^k, v_y^k, v_z^k, \rho^k, h^k)_{k=1, n}$

$G_{a_x} = \frac{v_x - v_x^{(n-1)}}{\Delta t} - (1 - \theta) a_x^{(n-1)} - \theta \left[- \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial x} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial x} \right) + a_{visco}^x + a_{grav}^x \right] \quad (2.170)$	
$G_{a_y} = \frac{v_y - v_y^{(n-1)}}{\Delta t} - (1 - \theta) a_y^{(n-1)} - \theta \left[- \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial y} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial y} \right) + a_{visco}^y + a_{grav}^y \right] \quad (2.171)$	
$G_{a_z} = \frac{v_z - v_z^{(n-1)}}{\Delta t} - (1 - \theta) a_z^{(n-1)} - \theta \left[- \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial z} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial z} \right) + a_{visco}^z + a_{grav}^z \right] \quad (2.172)$	
$G_{\dot{U}} = \frac{U - U^{(n-1)}}{\Delta t} - (1 - \theta) \dot{U}^{(n-1)} - \theta \left[\frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{U}_{visco} + \dot{U}_{cond} \right] \quad (2.173)$	
$G_{\dot{T}} = \frac{T - T^{(n-1)}}{\Delta t} - (1 - \theta) \dot{T}^{(n-1)} - \theta \left[\frac{1}{\Omega_i \rho_i^2} \cdot \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \cdot \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{T}_{visco} + \dot{T}_{cond} \right] \quad (2.174)$	
$G_{v_x} = \frac{x - x^{(n-1)}}{\Delta t} - (1 - \theta) v_x^{(n-1)} - \theta [v_x] \quad (2.175)$	
$G_{v_y} = \frac{y - y^{(n-1)}}{\Delta t} - (1 - \theta) v_y^{(n-1)} - \theta [v_y] \quad (2.176)$	
$G_{v_z} = \frac{z - z^{(n-1)}}{\Delta t} - (1 - \theta) v_z^{(n-1)} - \theta [v_z] \quad (2.177)$	
$G_{\dot{\rho}} = \frac{\rho - \rho^{(n-1)}}{\Delta t} - (1 - \theta) \dot{\rho}^{(n-1)} - \theta \left[\sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) \right] \quad (2.178)$	
$G_{\rho} = \rho - \theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j W_{ij}(h_i) \right] \quad (2.179)$	
$G_h = h - \theta \left[\frac{1}{\theta} \left(\frac{K}{\rho} \right)^{1/d} \right] \quad (2.180)$	

Cuadro 2.10: Esquema de todas las ecuaciones implícitas del *Newton-Raphson*. En cada sección podemos observar las ecuaciones relacionadas según las variables independientes que intervienen. Suponemos que $\Omega_i = cte$ y por aproximamos sus derivadas a 0. Los términos extra relativos a la gravedad son a_{grav}^x , a_{grav}^y y a_{grav}^z y se encuentra descritos en el apéndice C.2. Los términos extra relativos a la viscosidad son a_{visco}^x , a_{visco}^y , a_{visco}^z , \dot{U}_{visco} y \dot{T}_{visco} y se encuentra descritos en el apéndice C.1. Los términos extra relativos a la conducción térmica son \dot{U}_{cond} y \dot{T}_{cond} y se encuentra descritos en el apéndice C.3.

Sistema de ecuaciones en formato matricial Para utilizar las bibliotecas algorítmicas se necesitan matrices cuadradas, es decir, el mismo número de ecuaciones que de variables independientes. Si suponemos n_{vi} variables independientes, necesitaremos n_e ecuaciones por

partícula y suponiendo n partículas obtenemos las $n_e \cdot n$ ecuaciones G diferentes de todas las partículas, tal que si montamos el sistema de ecuaciones de forma matricial, donde J_G es la matriz jacobiana de las G' s, $\delta\chi$ es el vector de incógnitas y G es el vector de G' s obtenemos lo siguiente,

$$J_G \cdot \delta\chi = -G \quad (2.182)$$

El cuadro 2.11 muestra un ejemplo con un sistema de una única partícula ($n = 1$) con nueve variables independientes ($n_{vi} = 9$) y nueve ecuaciones ($n_e = 9$), en una iteración cualquiera.

$$\begin{pmatrix} \frac{\partial G_{ax}}{\partial x} & \frac{\partial G_{ax}}{\partial y} & \frac{\partial G_{ax}}{\partial z} & \frac{\partial G_{ax}}{\partial [U/T]} & \frac{\partial G_{ax}}{\partial v_x} & \frac{\partial G_{ax}}{\partial v_y} & \frac{\partial G_{ax}}{\partial v_z} & \frac{\partial G_{ax}}{\partial \rho} & \frac{\partial G_{ax}}{\partial h} \\ \frac{\partial G_{ay}}{\partial x} & \frac{\partial G_{ay}}{\partial y} & \frac{\partial G_{ay}}{\partial z} & \frac{\partial G_{ay}}{\partial [U/T]} & \frac{\partial G_{ay}}{\partial v_x} & \frac{\partial G_{ay}}{\partial v_y} & \frac{\partial G_{ay}}{\partial v_z} & \frac{\partial G_{ay}}{\partial \rho} & \frac{\partial G_{ay}}{\partial h} \\ \frac{\partial G_{az}}{\partial x} & \frac{\partial G_{az}}{\partial y} & \frac{\partial G_{az}}{\partial z} & \frac{\partial G_{az}}{\partial [U/T]} & \frac{\partial G_{az}}{\partial v_x} & \frac{\partial G_{az}}{\partial v_y} & \frac{\partial G_{az}}{\partial v_z} & \frac{\partial G_{az}}{\partial \rho} & \frac{\partial G_{az}}{\partial h} \\ \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial x} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial y} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial z} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial [U/T]} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_x} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_y} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_z} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial \rho} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial h} \\ \frac{\partial G_{vx}}{\partial x} & \frac{\partial G_{vx}}{\partial y} & \frac{\partial G_{vx}}{\partial z} & \frac{\partial G_{vx}}{\partial [U/T]} & \frac{\partial G_{vx}}{\partial v_x} & \frac{\partial G_{vx}}{\partial v_y} & \frac{\partial G_{vx}}{\partial v_z} & \frac{\partial G_{vx}}{\partial \rho} & \frac{\partial G_{vx}}{\partial h} \\ \frac{\partial G_{vy}}{\partial x} & \frac{\partial G_{vy}}{\partial y} & \frac{\partial G_{vy}}{\partial z} & \frac{\partial G_{vy}}{\partial [U/T]} & \frac{\partial G_{vy}}{\partial v_x} & \frac{\partial G_{vy}}{\partial v_y} & \frac{\partial G_{vy}}{\partial v_z} & \frac{\partial G_{vy}}{\partial \rho} & \frac{\partial G_{vy}}{\partial h} \\ \frac{\partial G_{vz}}{\partial x} & \frac{\partial G_{vz}}{\partial y} & \frac{\partial G_{vz}}{\partial z} & \frac{\partial G_{vz}}{\partial [U/T]} & \frac{\partial G_{vz}}{\partial v_x} & \frac{\partial G_{vz}}{\partial v_y} & \frac{\partial G_{vz}}{\partial v_z} & \frac{\partial G_{vz}}{\partial \rho} & \frac{\partial G_{vz}}{\partial h} \\ \frac{\partial G_{\rho}}{\partial x} & \frac{\partial G_{\rho}}{\partial y} & \frac{\partial G_{\rho}}{\partial z} & \frac{\partial G_{\rho}}{\partial [U/T]} & \frac{\partial G_{\rho}}{\partial v_x} & \frac{\partial G_{\rho}}{\partial v_y} & \frac{\partial G_{\rho}}{\partial v_z} & \frac{\partial G_{\rho}}{\partial \rho} & \frac{\partial G_{\rho}}{\partial h} \\ \frac{\partial G_h}{\partial x} & \frac{\partial G_h}{\partial y} & \frac{\partial G_h}{\partial z} & \frac{\partial G_h}{\partial [U/T]} & \frac{\partial G_h}{\partial v_x} & \frac{\partial G_h}{\partial v_y} & \frac{\partial G_h}{\partial v_z} & \frac{\partial G_h}{\partial \rho} & \frac{\partial G_h}{\partial h} \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta[U/T] \\ \delta v_x \\ \delta v_y \\ \delta v_z \\ \delta \rho \\ \delta h \end{pmatrix} = - \begin{pmatrix} G_{ax} \\ G_{ay} \\ G_{az} \\ G_{[\dot{U}/\dot{T}]} \\ G_{vx} \\ G_{vy} \\ G_{vz} \\ G_{\rho} \\ G_h \end{pmatrix}$$

Cuadro 2.11: Desarrollo del esquema implícito general para 1 partícula con 8 ecuaciones y 8 variables independientes.

Resolución del sistema de ecuaciones El resultado que se quiere obtener es el vector de incógnitas $\delta\chi$, y por tanto si cogemos la ecuación 2.183 y despejamos nos aparece la inversión de la matriz que puede ser resuelta directamente mediante cualquiera de las técnicas de inversión de matrices conocidas.

$$\delta\chi = (J_G)^{-1} \cdot -G \quad (2.183)$$

La solución anterior es eficaz pero no eficiente, ya que, la matriz jacobiana del sistema de ecuaciones es una matriz dispersa (contiene un número significativo de ceros en los coeficientes). En el problema que nos ocupa, la tasa de relleno es decreciente con el aumento del número de partículas y la disminución del número de vecinos objetivo. Por ejemplo, el valor de relleno de la matriz es sobre un 2 % para 10.000 partículas y un número de vecinos por partícula sobre 50.

Por tanto es mucho más inteligente utilizar un algoritmo especializado en matrices dispersas para resolver este problema. Además la forma de la matriz influye directamente en el algoritmo escogido, dependiendo de la forma de la dispersión, sera más o menos eficiente. En este caso, debemos saber que tenemos una matriz dispersa general (sin forma definida) y no simétrica (el elemento $i - j$ no se corresponde con el $j - i$). Existen en el mercado multitud de bibliotecas estandarizadas que aplican algoritmos paralelos para matrices dispersas y resuelven, de forma transparente al usuario, el sistema de ecuaciones obteniendo las $\delta\chi$ (véase sección 3.3.2).

Agrupaciones de ecuaciones en bloques Debido a que la matriz es dispersa una elección esencial es el orden en el que se colocan las ecuaciones en el sistema, ya que esto determina la forma de la dispersión. La intención siempre debe ser intentar que la matriz dispersa sea lo más diagonal posible y para ello se toma una serie de criterios. La dispersión esta relacionada con las partículas y sus vecinas, para evitar añadir dispersión extra a la matriz, es necesario que las ecuaciones de cada partícula permanezcan juntas en filas consecutivas. Además, por la misma razón es necesario que las interacciones con las otras partículas j permanezcan en columnas consecutivas. De esta forma tenemos la matriz seccionada en conjuntos de filas pertenecientes a las ecuaciones de cada partícula y en conjunto de columnas también pertenecientes a las variables independientes de cada una de las partículas, formando bloques de coeficientes en la matriz jacobiana, diferenciados por partículas. El orden interno de filas y columnas dentro de los bloques también puede afectar a la dispersión, pero en un porcentaje despreciable frente a la agrupación en bloques. Estos bloques, según sus interacciones, se pueden clasificar en tres tipos.

- Bloques de partícula i : Situados en la diagonal de la matriz. Si observamos las ecuaciones contienen los sumatorios de las contribuciones de sus partículas vecinas.
- Bloques de vecinos $i - j$ (con $i \neq j$ y j vecina de i): Situados fuera de la diagonal, pero en el conjunto de filas pertenecientes a la partícula i . Al derivar las ecuaciones de i respecto de las variables independientes de j desaparecen los sumatorios y solo sobrevive un único término que es la contribución directa que hace j sobre i .
- Bloques nulos $i - j$ (con $i \neq j$ y j no vecina de i) Situados fuera de la diagonal, pero en el conjunto de filas pertenecientes a la partícula i . Al derivar las ecuaciones de i respecto de las variables independientes de j toman valor cero.

Es necesario recalcar que la forma de colocar los coeficientes es algo determinante en la dispersión de la matriz y por tanto afecta directamente al coste de los algoritmos de resolución de matrices dispersas. Si no prestamos atención a este detalle cometeremos un gravísimo error porque esta parte determina el coste computacional de todo el programa (a partir de un número de partículas pequeño, por ejemplo 10.000, mucho menor que el número de partículas habitual en los códigos explícitos que es de decenas de miles hasta millones de partículas). De nada sirve reducir el tiempo de calculo usando arboles de búsqueda de vecinos o aproximaciones del potencial gravitatorio si luego no resolvemos de forma eficiente el sistema de ecuaciones.

Convergencia del esquema *Newton-Raphson* en las ecuaciones Una vez resuelto el sistema de ecuaciones, se actualizan las variables y se comprueba la convergencia de las ecuaciones. El objetivo del *Newton-Raphson* es encontrar los valores de las variables independientes que, tras una determinada iteración n , anulan todas las ecuaciones o, en el esquema numérico propuesto, los valores se aproximan a la solución en una distancia menor o igual a un cierto error numérico permitido (ε). Con los sucesivos refinamientos el resultado se va aproximando cada vez más a cero.

$$\begin{aligned} \text{Convergencia} = & (G_{a_x} \leq \varepsilon_{a_x}) \wedge (G_{a_y} \leq \varepsilon_{a_y}) \wedge (G_{a_z} \leq \varepsilon_{a_z}) \\ & \wedge (G_{[\dot{U}/\dot{T}]} \leq \varepsilon_{[\dot{U}/\dot{T}]}) \\ & \wedge (G_{v_x} \leq \varepsilon_{v_x}) \wedge (G_{v_y} \leq \varepsilon_{v_y}) \wedge (G_{v_z} \leq \varepsilon_{v_z}) \\ & \wedge (G_{\rho} \leq \varepsilon_{\rho}) \end{aligned} \tag{2.184}$$

$$\wedge (G_h \leq \varepsilon_h) \tag{2.185}$$

Como es un sistema de ecuaciones *Newton-Raphson* acoplado, se quiere obtener una precisión conjunta mayor y por tanto se establecen condiciones de convergencia individuales más restrictivas en las ecuaciones, que la restricción estándar de los métodos *Newton-Rapson* aplicada en las variables, es decir, si cada una de las ecuaciones iG_j se puede descomponer en sumas, tal que ${}^iG_j = ({}^iG_j)_1 + ({}^iG_j)_2 + \dots + ({}^iG_j)_n$. Consideramos estable cada una de las ecuaciones si, una vez actualizadas las variables, cumplen el criterio siguiente, donde ε se coloca en el rango de $[10^{-6}, 10^{-1}]$ siendo habitual el valor $\varepsilon \leq 10^{-4}$ pero debe depender de la ecuación y el modelo concreto que se trata.

$$\frac{|{}^iG_j|}{\sqrt{({}^iG_j)_1^2 + ({}^iG_j)_2^2 + \dots + ({}^iG_j)_n^2}} \leq \varepsilon \tag{2.186}$$

Las ecuaciones de convergencia completas podemos consultarlas en el apéndice [E](#). Como

ejemplo se muestra la ecuación de convergencia G_{a_x} , .

$$\frac{\left| \frac{v_x - v_x^{(n-1)}}{\Delta t} - (1 - \theta) a_x^{(n-1)} - \theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial x} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial x} \right) \right] - \theta a_{visc}^x \right|}{\sqrt{\left(\frac{v_x - v_x^{(n-1)}}{\Delta t} \right)^2 + \left(-(1 - \theta) a_x^{(n-1)} \right)^2 + \left(-\theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial x} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial x} \right) \right] \right)^2}} + (2.187)$$

Este criterio de convergencia en las ecuaciones tiene el defecto de que si la ecuación tiene un denominador prácticamente cero y la ecuación pertenece a una partícula quasi-estática, puede introducir errores artificiales de convergencia que limiten el paso de tiempo. Si establecemos límites estritos, en algunos modelos, podremos observar que las ecuaciones que controlan el paso del tiempo pertenecen a partículas que no resultan de interés, ni tienen ninguna afectación sobre lo que se pretende simular.

Convergencia del esquema *Newton-Raphson* en las correcciones Para evitar el defecto del criterio de convergencia en ecuaciones podemos establecer el criterio de convergencia en función de las correcciones normalizadas del Newton-Raphson. Este método consiste en dar la iteración por correcta cuando todas las correcciones en las variables independientes, de un refinamiento a otro, se situen por debajo de una cierta cota ($\varepsilon = 10^{-4}$).

$$\left| \frac{\delta \chi_i}{\chi_i} \right| \leq \varepsilon \quad (2.188)$$

Si tenemos dentro del esquema variables independientes de diferentes tipos (por ejemplo densidad y velocidad), es posible controlar solo uno o varios tipos de estas variables.

Como en el caso anterior, también es posible permitir un cierto porcentaje de variables independientes no controladas.

Reordenación de partículas En nuestro algoritmo, el orden de partículas determina como se monta la matriz. Como se nombró anteriormente esto afecta directamente a la dispersión. Por tanto es muy recomendable, a cada paso de tiempo de evolución reordenar el vector de partículas de tal forma que, estas se sitúen próximas espacialmente a sus vecinas, o al menos en la medida de lo posible. Un buen sistema es la ordenación *z-order* basada en el método de *Morton, 1966* [64], se puede observar la distribución en 3D en la figura 2.9.

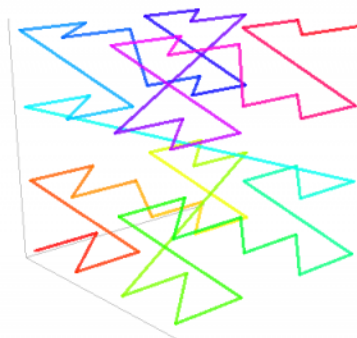


Figura 2.9: Distribución espacial en 3D realizada por el algoritmo z-order de Morton.

Variables independientes

Como se ha comentado anteriormente el sistema es modular en cuanto a las variables que intervienen en el cálculo, pudiendo intervenir unas u otras en función de las necesidades del modelo. Cada una de las variables las podemos utilizar según se describe en la siguiente lista:

- Posiciones(X, Y, Z) - (ecuaciones del momento [2.170](#), [2.171](#), [2.172](#)):
 1. *Variables independientes*: $1D(G_{ax})$, $2D(G_{ax}, G_{ay})$, $3D(G_{ax}, G_{ay}, G_{az})$. Según las dimensiones se incluyen las ecuaciones del momento necesarias en el esquema.
 2. *Magnitudes constantes*: No varían y no se incluyen en el esquema.
- Energía//Temperatura ($T//U$) - (ecuaciones de la energía [2.173](#) o ecuación de la temperatura [2.174](#)):
 1. *Variable independiente*: U ; Variable dependiente: T . G_U se incluye en el esquema y G_T no (actualizándose con la ecuación de estado).
 2. *Variable independiente*: T ; Variable dependiente: U . G_T se incluye en el esquema y G_U no (actualizándose con la ecuación de estado).
 3. *Magnitudes constantes*: No varían y no se incluyen en el esquema.
- Velocidades (Vx, Vy, Vz) - (ecuaciones de la velocidad [2.175](#), [2.176](#), [2.177](#)):
 1. *Variables independientes*: $1D(G_{vx})$, $2D(G_{vx}, G_{vy})$, $3D(G_{vx}, G_{vy}, G_{vz})$. Según las dimensiones se incluyen las ecuaciones del movimiento necesarias en el esquema.
 2. *Variables dependientes*: $1D(v_x)$, $2D(v_x, v_y)$, $3D(v_x, v_y, v_z)$; Las variaciones en velocidad se reducen a las variaciones en posición (ecuación [2.189](#)) y se agregan al resto de ecuaciones.
 3. *Variables nulas*: Valen cero y no se incluyen en el esquema.
- Densidad (ρ) - (ecuaciones de la continuidad [2.178](#) o de la densidad [2.179](#)):
 1. *Variable independiente*: $G_{\dot{\rho}}$. Se incluye la ecuación de la continuidad en el esquema.

2. *Variable independiente:* G_ρ . Se incluye la ecuación de la densidad.
 3. *Variable dependiente:* Las variaciones en densidad se reducen a las variaciones en posición (ecuación 2.190) y se agregan al resto de ecuaciones del esquema.
 4. *Variable dependiente:* ρ se recalcula al inicio de cada iteración, de forma exacta, según la ecuación 2.61.
- Distancia de interpolacion (h) - (ecuacion de la distancia de interpolación 2.180) :
1. *Variable independiente:* G_h . Se incluye la ecuación de la relación entre densidad y volumen en el esquema.
 2. *Variable dependiente:* La variable se calcula a cada iteración estable.
 3. *Magnitud constante:* No varía y no se incluye en el esquema

Cabe destacar, que es posible sustituir las variaciones en velocidad ($\delta v_x, \delta v_y, \delta v_z$), las variaciones en densidad ($\delta \rho$) y las variaciones de la distancia de interpolacion(δh), por variaciones en las posiciones ($\delta x, \delta y, \delta z$), con el fin de reducir el número de ecuaciones incluidas en el esquema y así aumentar la velocidad de resolución del sistema de ecuaciones.

Desgraciadamente esto no afecta al coste computacional (que se sigue situando en $\Theta(n^{2-3})$, es decir entre cuadrático y cúbico, dependiendo del método de resolución de la matriz), pero si reduce en gran medida el tiempo de computación (porque el tamaño de la entrada n , es menor). Al disminuir la relación variables-ecuaciones disminuye el tamaño de la matriz y el número de ceros, pero esto no afecta al coste algorítmico, solo al tiempo de calculo, al reducir el tamaño del problema.

Reducción de la variable independiente velocidad (\mathbf{v}) a una variable dependiente:

Utilizando la ecuación de la velocidad 2.56 podemos obtener las variaciones en las diferentes dimensiones, tal que $v_k(k)$.

$$\delta v_x = \frac{2}{\Delta t} \cdot \delta x \quad ; \quad \delta v_y = \frac{2}{\Delta t} \cdot \delta y \quad ; \quad \delta v_z = \frac{2}{\Delta t} \cdot \delta z \quad (2.189)$$

Reducción de la variable independiente densidad (ρ) a una variable dependiente:

Utilizando la ecuación de la densidad 2.61 podemos obtener las variaciones en las diferentes dimensiones, tal que $\rho(x, y, z, h)$.

$$\delta \rho = \sum_{j=1}^n m_j \frac{\partial W_{ij}(h_i)}{\partial x} \cdot \delta x + \sum_{j=1}^n m_j \frac{\partial W_{ij}(h_i)}{\partial y} \cdot \delta y + \sum_{j=1}^n m_j \frac{\partial W_{ij}(h_i)}{\partial z} \cdot \delta z + \sum_{j=1}^n m_j \frac{\partial W_{ij}(h_i)}{\partial h} \cdot \delta h \quad (2.190)$$

Linealización de las ecuaciones reducidas En contraposición a la linealización estándar (ecuación 2.181), en la linealización reducida se eliminan del esquema de variación las variables independientes velocidad y densidad obteniendo la siguiente variación de las ecuaciones (δG).

$$\begin{aligned}
 \delta G = & \left[\frac{\partial G}{\partial x} + \frac{\partial G}{\partial v_x} \cdot \frac{1}{\Delta t} + \frac{\partial G}{\partial \rho} \cdot \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x} \right] \cdot \delta x \\
 & + \left[\frac{\partial G}{\partial y} + \frac{\partial G}{\partial v_y} \cdot \frac{1}{\Delta t} + \frac{\partial G}{\partial \rho} \cdot \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y} \right] \cdot \delta y \\
 & + \left[\frac{\partial G}{\partial z} + \frac{\partial G}{\partial v_z} \cdot \frac{1}{\Delta t} + \frac{\partial G}{\partial \rho} \cdot \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z} \right] \cdot \delta z \\
 & + \left[\frac{\partial G}{\partial [U/T]} \right] \cdot \delta [U/T] \\
 & + \left[\frac{\partial G}{\partial h} + \frac{\partial G}{\partial \rho} \cdot \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h} \right] \cdot \delta h = -G
 \end{aligned} \tag{2.191}$$

Donde $[U/T]$, es la elección de un esquema con la ecuación de la energía o con la ecuación de la temperatura.

Esquemas de trabajo especiales

En los siguientes apartados se muestran esquemas de trabajo basados en la reducción de las variables. Estos esquemas pueden ser utilizados en los modelos específicos que se verán en los tests básicos (capítulo 4) y los tests con gravedad (capítulo 5). Como en el ejemplo que se muestra en el caso general (cuadro 2.11), por simplicidad, supondremos para los siguientes ejemplos un desarrollo del esquema semi-implícito general para 1 partícula con n_v variables independientes y coincidiendo con n_e ecuaciones.

1) Esquema SPH implícito: Variación solo en energía o temperatura En el test de la expansión de una onda térmica (sección 4.5), las partículas permanecen inmóviles y suponemos que no hay transferencia de masa entre las partículas, entonces obtenemos un esquema de trabajo con una matriz 1×1 con las mismas ecuaciones que variables independientes.

$$\left(\frac{\partial G_{[U/T]}}{\partial [U/T]} \right) \cdot \left(\delta [U/T] \right) = - \left(G_{[U/T]} \right)$$

2) Esquema SPH implícito: Variación en posiciones Si diseñamos modelos isotermos en 3D y suponemos que no hay transferencia de masa entre las partículas, entonces obtenemos un esquema de trabajo con una matriz $m \times m$ donde $m = n \cdot n_v$, siendo n es número de partículas y $n_v = 4$ el número de variables que es igual a n_e o número de ecuaciones. En este caso el tamaño de bloque es de 2×2 a 4×4 ($n_e \times n_v$), en función del número de dimensiones.

$$\begin{pmatrix} \frac{\partial G_{ax}}{\partial x} & \frac{\partial G_{ax}}{\partial y} & \frac{\partial G_{ax}}{\partial z} & \frac{\partial G_{ax}}{\partial h} \\ \frac{\partial G_{ay}}{\partial x} & \frac{\partial G_{ay}}{\partial y} & \frac{\partial G_{ay}}{\partial z} & \frac{\partial G_{ay}}{\partial h} \\ \frac{\partial G_{az}}{\partial x} & \frac{\partial G_{az}}{\partial y} & \frac{\partial G_{az}}{\partial z} & \frac{\partial G_{az}}{\partial h} \\ \frac{\partial G_h}{\partial x} & \frac{\partial G_h}{\partial y} & \frac{\partial G_h}{\partial z} & \frac{\partial G_h}{\partial h} \end{pmatrix} \cdot \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta h \end{pmatrix} = - \begin{pmatrix} G_{ax} \\ G_{ay} \\ G_{az} \\ G_h \end{pmatrix}$$

3) Esquema SPH implícito general reducido Si diseñamos modelos generales en 3D aplicando el esquema reducido, entonces obtenemos un esquema de trabajo con una matriz $m \times m$ donde $m = n \cdot n_v$, siendo n es número de partículas y n_v el número de variables que es igual a n_e o número de ecuaciones. En este caso el tamaño de bloque depende de las variables independientes extra pudiendo ser de 3×3 a 5×5 ($n_e \times n_v$), donde n_v depende de si incluimos o no energía/temperatura al modelo.

- Variación en posiciones (x, y, z), energía (U) y distancia de interpolacion (h).
- Variación en posiciones (x, y, z), temperatura (T) y distancia de interpolacion (h).

$$\begin{pmatrix} \frac{\partial G_{ax}}{\partial x} & \frac{\partial G_{ax}}{\partial y} & \frac{\partial G_{ax}}{\partial z} & \frac{\partial G_{ax}}{\partial [U/T]} & \frac{\partial G_{ax}}{\partial h} \\ \frac{\partial G_{ay}}{\partial x} & \frac{\partial G_{ay}}{\partial y} & \frac{\partial G_{ay}}{\partial z} & \frac{\partial G_{ay}}{\partial [U/T]} & \frac{\partial G_{ay}}{\partial h} \\ \frac{\partial G_{az}}{\partial x} & \frac{\partial G_{az}}{\partial y} & \frac{\partial G_{az}}{\partial z} & \frac{\partial G_{az}}{\partial [U/T]} & \frac{\partial G_{az}}{\partial h} \\ \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial x} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial y} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial z} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial [U/T]} & \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial h} \\ \frac{\partial G_h}{\partial x} & \frac{\partial G_h}{\partial y} & \frac{\partial G_h}{\partial z} & \frac{\partial G_h}{\partial [U/T]} & \frac{\partial G_h}{\partial h} \end{pmatrix} \cdot \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta [U/T] \\ \delta h \end{pmatrix} = - \begin{pmatrix} G_{ax} \\ G_{ay} \\ G_{az} \\ G_{[\dot{U}/\dot{T}]} \\ G_h \end{pmatrix}$$

En el cuadro 2.12 se muestra el valor de los coeficientes de la matriz del esquema reducido del ejemplo 3), siguiendo el desarrollo de la ecuación 2.191.

$\frac{\partial G_{a_k}}{\partial x}$	$\frac{\partial G_{a_k}}{\partial x} + \frac{1}{\Delta t} \frac{\partial G_{a_k}}{\partial v_x} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_{a_k}}{\partial y}$	$\frac{\partial G_{a_k}}{\partial y} + \frac{1}{\Delta t} \frac{\partial G_{a_k}}{\partial v_y} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_{a_k}}{\partial z}$	$\frac{\partial G_{a_k}}{\partial z} + \frac{1}{\Delta t} \frac{\partial G_{a_k}}{\partial v_z} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_{a_k}}{\partial [U/T]}$	$\frac{\partial G_{a_k}}{\partial [U/T]}$
$\frac{\partial G_{a_k}}{\partial h}$	$\frac{\partial G_{a_k}}{\partial h} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$
$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial x}$	$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial x} + \frac{1}{\Delta t} \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_x} + \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial y}$	$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial y} + \frac{1}{\Delta t} \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_y} + \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial z}$	$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial z} + \frac{1}{\Delta t} \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial v_z} + \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial [U/T]}$	$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial [U/T]}$
$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial h}$	$\frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial h} + \frac{\partial G_{[\dot{U}/\dot{T}]}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$
$\frac{\partial G_h}{\partial x}$	$\frac{\partial G_h}{\partial x} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_h}{\partial y}$	$\frac{\partial G_h}{\partial y} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_h}{\partial z}$	$\frac{\partial G_h}{\partial z} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_h}{\partial [U/T]}$	$\frac{\partial G_h}{\partial [U/T]}$
$\frac{\partial G_h}{\partial h}$	$\frac{\partial G_h}{\partial h} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$

Cuadro 2.12: Coeficientes de la matriz del esquema 3) reducido en velocidad y densidad. Donde $k = x, y, z$

4) Esquema SPH con aproximación implícita quasi-hidrostática Si suponemos un modelo en equilibrio, es decir con velocidad nula (y por tanto no se aplica la reducción de la variable \mathbf{v}), con reducción en densidad, pero con variación en energía/temperatura. El esquema de de trabajo es una matriz $m \times m$ donde $m = n \cdot n_v$, siendo n es número de partículas y $n_v = 5$ el número de variables (en 3D). En este caso el tamaño de bloque es 5×5 ($n_v \times n_v$). El equilibrio de las ecuaciones se consigue mediante la ecuación del momento modificada (el término de la velocidad es nulo) y la ecuación de la energía modificada según el siguiente esquema:

$$\mathbf{a} = 0 = - \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \nabla_i \mathbf{W}_{ij}(h_i) + \frac{P_j}{\Omega_j \rho_j^2} \nabla_i \mathbf{W}_{ij}(h_j) \right) \quad (2.192)$$

$$\dot{U} = \frac{d U_i}{d t} = \frac{P_i}{\Omega_i \rho_i^2} \cdot (\rho_i - \rho_i^{(n-1)}) \quad (2.193)$$

$$\rho = \sum_{j=1}^N m_j W_{ij}(h_i) \quad (2.194)$$

$$h = \left(\frac{K}{\rho} \right)^{1/d} \quad (2.195)$$

Cuadro 2.13: Discretización de las ecuaciones del SPH: Esquema cuasi-hidrostático

$$\begin{pmatrix} \frac{\partial G_{ax}}{\partial x} & \frac{\partial G_{ax}}{\partial y} & \frac{\partial G_{ax}}{\partial z} & \frac{\partial G_{ax}}{\partial U} & \frac{\partial G_{ax}}{\partial h} \\ \frac{\partial G_{ay}}{\partial x} & \frac{\partial G_{ay}}{\partial y} & \frac{\partial G_{ay}}{\partial z} & \frac{\partial G_{ay}}{\partial U} & \frac{\partial G_{ay}}{\partial h} \\ \frac{\partial G_{az}}{\partial x} & \frac{\partial G_{az}}{\partial y} & \frac{\partial G_{az}}{\partial z} & \frac{\partial G_{az}}{\partial U} & \frac{\partial G_{az}}{\partial h} \\ \frac{\partial G_{\dot{U}}}{\partial x} & \frac{\partial G_{\dot{U}}}{\partial y} & \frac{\partial G_{\dot{U}}}{\partial z} & \frac{\partial G_{\dot{U}}}{\partial U} & \frac{\partial G_{\dot{U}}}{\partial h} \\ \frac{\partial G_h}{\partial x} & \frac{\partial G_h}{\partial y} & \frac{\partial G_h}{\partial z} & \frac{\partial G_h}{\partial U} & \frac{\partial G_h}{\partial h} \end{pmatrix} \cdot \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta U \\ \delta h \end{pmatrix} = - \begin{pmatrix} G_{ax} \\ G_{ay} \\ G_{az} \\ G_{\dot{U}} \\ G_h \end{pmatrix}$$

Este esquema permite utilizar pasos de tiempo muy altos ($\Delta t \uparrow$), ya que la escala activa es infinita en Δx al no moverse las partículas.

Aplicando ahora solamente la reducción en densidad, pero no en velocidad se muestra el valor de los coeficientes de la matriz del esquema reducido (cuadro 2.14).

$\frac{\partial G_{a_k}}{\partial x}$	$\frac{\partial G_{a_k}}{\partial x} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_{a_k}}{\partial y}$	$\frac{\partial G_{a_k}}{\partial y} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_{a_k}}{\partial z}$	$\frac{\partial G_{a_k}}{\partial z} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_{a_k}}{\partial U}$	$\frac{\partial G_{a_k}}{\partial U}$
$\frac{\partial G_{a_k}}{\partial h}$	$\frac{\partial G_{a_k}}{\partial h} + \frac{\partial G_{a_k}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$
$\frac{\partial G_{\dot{U}}}{\partial x}$	$\frac{\partial G_{\dot{U}}}{\partial x} + \frac{\partial G_{\dot{U}}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_{\dot{U}}}{\partial y}$	$\frac{\partial G_{\dot{U}}}{\partial y} + \frac{\partial G_{\dot{U}}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_{\dot{U}}}{\partial z}$	$\frac{\partial G_{\dot{U}}}{\partial z} + \frac{\partial G_{\dot{U}}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_{\dot{U}}}{\partial U}$	$\frac{\partial G_{\dot{U}}}{\partial U}$
$\frac{\partial G_{\dot{U}}}{\partial h}$	$\frac{\partial G_{\dot{U}}}{\partial h} + \frac{\partial G_{\dot{U}}}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$
$\frac{\partial G_h}{\partial x}$	$\frac{\partial G_h}{\partial x} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x}$
$\frac{\partial G_h}{\partial y}$	$\frac{\partial G_h}{\partial y} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y}$
$\frac{\partial G_h}{\partial z}$	$\frac{\partial G_h}{\partial z} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z}$
$\frac{\partial G_h}{\partial U}$	$\frac{\partial G_h}{\partial U}$
$\frac{\partial G_h}{\partial h}$	$\frac{\partial G_h}{\partial h} + \frac{\partial G_h}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h}$

Cuadro 2.14: Coeficientes de la matriz del esquema 4) reducido en densidad con $k = x, y, z$

5) Esquema SPH implícito general Si diseñamos modelos generales en 3D entonces obtenemos un esquema de trabajo con una matriz $m \times m$ donde $m = n \cdot n_v$, siendo n es número de partículas y n_v el número de variables. En este caso el tamaño de bloque depende de las variables independientes escogidas (n_v) puede ser de 7×7 a 9×9 , dependiendo de las las dimensiones. Este es el caso general, ya expuesto en el cuadro 2.11.

Capítulo 3

Implicit SPH For Astrophysical Applications (ISFAA)

En esta tesis se presenta un código SPH implícito partiendo de la experiencia de nuestro grupo de investigación en las versiones SPH explícitas y en trabajos previos sobre hidrodinámica SPH implícita como el de *Knapp, 2000* [47].

Los pasos realizados antes de la publicación de este documento han dado lugar a un código implícito SPH que cumple con los requisitos descritos en la introducción, es capaz de pasar con éxito los tests (capítulo 4) y es aplicado a varios escenarios astrofísicos (capítulo 5).

Avances en la física del nuevo código implícito

El código incluye, como gran novedad, la base de las ecuaciones de Euler-Lagrange (basadas en el principio variacional), y representa los siguientes avances en relación al código SPH implícito que se describe en *Knapp, 2000* [47]:

- *Paralelización del código:* El coste computacional del código propuesto y el número de partículas objetivo, necesitan de la paralelización para conseguir resultados en tiempos aceptables. La paralelización es un requisito obligatorio para conseguir la efectividad de la implementación implícita del SPH.
- *Bibliotecas de resolución de sistemas de ecuaciones en paralelo:* La gestión de la paralelización de la parte del código relacionada con la inversión de la matriz es derivada a bibliotecas estándar de algoritmos de resolución de sistemas de ecuaciones.

- *Ecuación de estado (EOS)*: La EOS de gas ideal se utiliza para los tests básicos. Además, se han realizado simulaciones usando una EOS más realista que cuenta con las contribuciones de los iones, las correcciones coulombianas, los electrones degenerados y la radiación, siendo por tanto, más adecuada para el tratamiento de las condiciones existentes en los interiores estelares.
- *Gravedad*: La inclusión de la gravedad es obligatoria en muchos problemas astrofísicos. Se han realizado tres implementaciones diferentes que pueden ser utilizadas en función del modelo a tratar (sección 2.2.4). Este punto no fue tratado en el trabajo de *Knapp, 2000* [47].
- *Viscosidad artificial*: Se implementa este ingrediente básico para el tratamiento de choques en SPH, a diferencia de *Knapp* que lo cita pero no lo llega a implementar en el código (sección 2.2.5).
- *Conductividad térmica*: El flujo térmico de calor entre regiones de temperatura diferente es simulado gracias a la implementación de la conducción de calor (sección 2.2.6). Este método de disipación de calor ha sido utilizado en el test de onda térmica y puede utilizarse, por ejemplo, para la simulación de enanas blancas masivas.
- *Variable independiente temperatura*: La temperatura puede ser utilizada como variable independiente dentro del esquema calculando la energía mediante la EOS. Para ello se toma la misma definición que en *Lorén-Aguilar, 2009* [52]. Este método se debe utilizar en donde intervienen enanas blancas.
- *Variable independiente energía unido a la variación en temperatura*: Se implementa la variación en energía integrándola en el esquema. Además se actualiza la temperatura, mediante la EOS, a diferencia de *Knapp* que usa temperatura constante.
- *Variables reducidas*: Este es uno de los grandes avances del código. Es un mecanismo algebraico que permite poner las variables densidad y velocidad en función de las posiciones para reducir el tamaño de la matriz del sistema de ecuaciones (sección 2.3.4).
- *Interpolador tipo Sinc*: Más flexible que los polinomios estándar y que son más resistentes al *pairing-instability*.

3.1. Esquema de trabajo: Flujo de ejecución

La ejecución del programa se efectúa en cuatro grandes apartados:

1. Cargar el modelo de entrada: Necesitamos disponer de un modelo inicial donde se definan las posiciones de las partículas, velocidades, masas, etc. En algunos casos estos modelos se encuentran previamente relajados mediante códigos SPH explícitos.
2. Iterar: Es donde se realizará la evolución del modelo. En este apartado se ejecutarán sucesivos refinamientos de Newton-Raphson para conseguir el objetivo de la convergencia de la iteración. Conforme el modelo evoluciona se guardan a disco los resultados obtenidos.
3. Refinar: Contiene el tratamiento específico de cada refinamiento de Newton-Raphson. En este apartado se resuelve el sistema de ecuaciones y se calculan las aproximaciones a las variables. Y por último, una vez actualizadas las partículas, se comprueba la convergencia de las ecuaciones.
4. Finalizar la simulación: Ejecuta los scripts necesarios para analizar los resultados y crear las graficas y/o vídeos escogidos para el estudio del modelo.

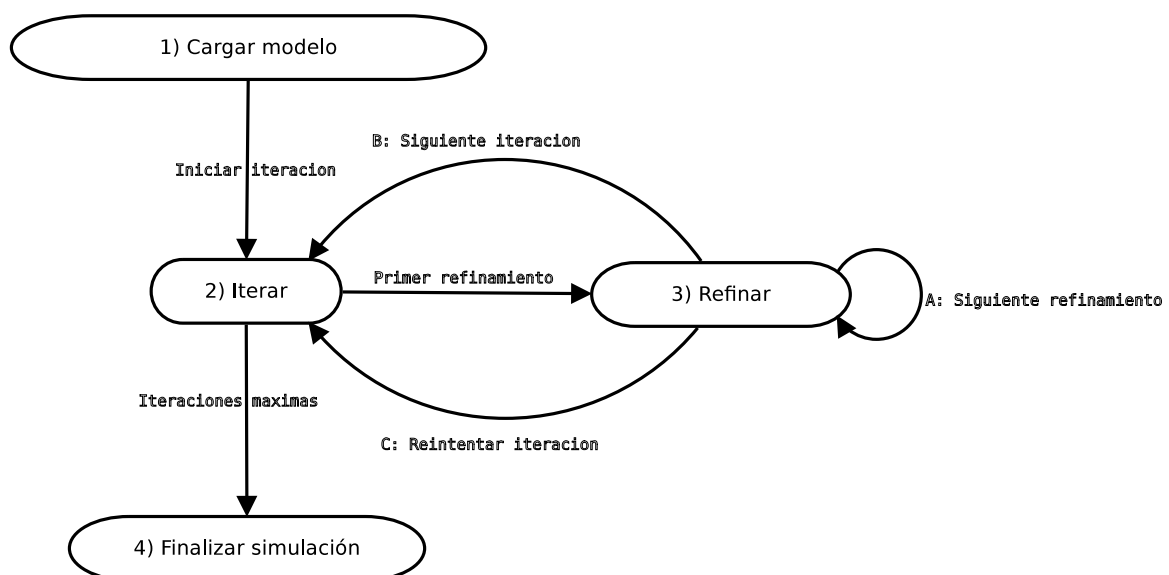


Figura 3.1: ISFAA: Diagrama de flujo de la secuencia de ejecución, donde se especifican las áreas de actuación del código.

3.1.1. Parametrización

El código se ha descrito de forma general y mediante unos parámetros se puede tunear según el modelo que se necesite tratar. Este es un listado de los parámetros más generales que se pueden especificar:

- El número de dimensiones del modelo. Una vez fijado y de forma automática, solamente las ecuaciones necesarias en para ese número de dimensiones son tenidas en cuenta.
- El tipo de esquema: 1) Explícito (primer orden), 2) Implícito (primer orden), 3) Semi-implícito o centrado (segundo orden).
- El tipo de ecuación de estado (EOS) utilizada: Ideal, politrópica o más realista.
- La selección de variables independientes que se tomarán en la simulación: Posiciones, velocidades, densidad, energía/temperatura.
- Los componentes físicos que se utilizarán en el modelo: Gravedad, viscosidad artificial, conducción térmica y/o red nuclear.
- El grado de paralelismo: El número de procesadores entre los que se generará la paralelización.
- El número de vecinos objetivo.
- Las condiciones límite de convergencia: Refinamientos máximos para probar cada paso de tiempo, condiciones de convergencia a aplicar, etc.
- Y más parámetros específicos de los modelos concretos.

Además, para evitar ejecutar instrucciones innecesarias que escojan entre los diferentes parámetros iniciales también se han implementado versiones específicas del código, con los parámetros previamente ya seleccionados, para tratar los problemas computacionalmente más complejos (básicamente los que necesitan de un número de partículas mayor para obtener los resultados).

3.2. Iteración

El objetivo de la iteración es aproximar todas las variables de cada partícula en un paso de tiempo determinado. Este proceso se realiza mediante refinamientos de las derivadas temporales aplicando un método Newton-Raphson. Una vez se ha finalizado cada refinamiento se comprueba la convergencia de las iteraciones, de tal modo que si ha tenido éxito se actualiza el modelo y se avanza a la siguiente iteración y si la iteración ha fracasado se reintenta la iteración con un paso de tiempo menor.

Además, dependiendo de los parámetros iniciales podemos establecer cada cuantas iteraciones estables se almacenarán los datos a disco para su posterior análisis. Como este procedimiento necesita la ejecución de comandos de entrada/salida que graben los datos al disco físico, con el consiguiente costo en tiempo de ejecución, puede ser interesante que este proceso se realice únicamente cada cierto número de iteraciones o cada cierto tiempo físico del modelo, en vez de cada iteración. Esta posibilidad está contemplada en el código ISFAA.

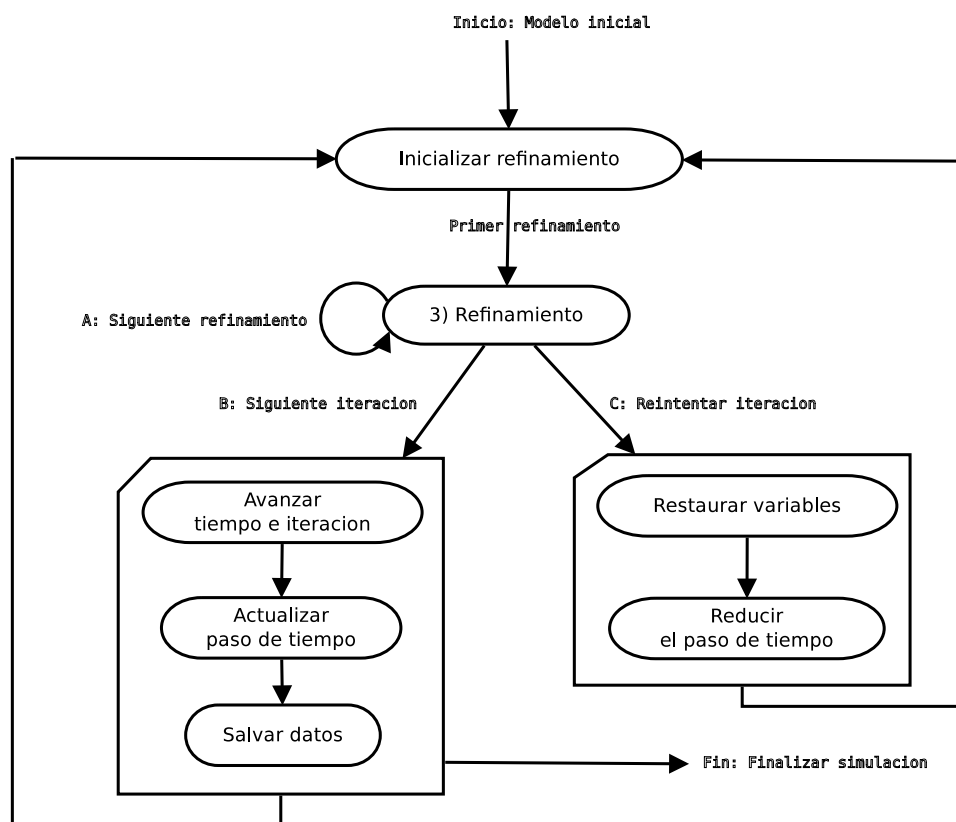


Figura 3.2: ISFAA: Diagrama de flujo de las iteraciones. Aquí se especifican, a grandes rasgos, la actuación del código en el área 2)Iterar de la figura 3.1.

Los datos de salida se descomponen en dos ficheros:

- Datos generales: Acumula los valores generales, para todas las partículas, de cada una de las iteraciones estables que se guardan a disco. Es decir, tiempo, paso de tiempo, energía total, etc. Este fichero es único para toda la simulación y permite crear gráficos que afecten a todas las iteraciones, como por ejemplo la evolución temporal del paso de tiempo. o la conservación de la energía.
- Datos específicos: Acumula los valores específicos por cada una de las partículas. Es decir, posiciones, velocidades, energía, temperatura, densidad, etc. Se crea un fichero diferente por cada iteración estable que se guarda a disco y permite efectuar vídeos de evolución del modelo a lo largo del tiempo, como por ejemplo la evolución temporal de la temperatura o los cambios de densidad en el sistema.

En caso de no convergencia, se restituyen los valores estables de la iteración anterior (que deben de estar almacenados) y se disminuye el paso de tiempo a la mitad. Después se reinician los parámetros que afectan a los refinamientos y reintenta la iteración.

3.2.1. Refinando el esquema Newton-Raphson

Como se ha explicado ampliamente en la sección 2.3.4, con el fin de determinar si una iteración es estable es necesario comprobar las correcciones en las ecuaciones o las correcciones en las variables al final del refinamiento. Cuando todas las correcciones en ecuaciones o correcciones en variables, cumplen con un error por debajo de un umbral crítico ε , la iteración se considera estable.

Además se realiza una comprobación adicional con el fin de evitar que la actualización imponga valores extraños (no físicos) como densidades, temperaturas o energías negativas. Si este es el caso la iteración se considera no convergente e inmediatamente se repite la iteración con un paso de tiempo menor.

3.2.2. Actualización del modelo

Dependiendo de los resultados del control de la convergencia se toman varias alternativas.

- Si *el refinamiento converge*, por tanto *la iteración converge*. Se almacenan las nuevas variables estables, se avanza la iteración, se reinician los refinamientos y se actualiza el

tiempo y el paso de tiempo.

$$iter = iter + 1 \quad ; \quad ref = 1 \quad ; \quad t = t^{(n-1)} + \Delta t \quad (3.1)$$

- Si *el refinamiento no converge* y no se ha llegado a los refinamientos máximos. Se prueba otro refinamiento.

$$ref = ref + 1 \quad (3.2)$$

- Si *el refinamiento no converge* y se ha llegado a los refinamientos máximos, *la iteración no converge*. Se restauran los valores iniciales de la iteración y se reduce el paso de tiempo.

$$\Delta t = \frac{\Delta t^{(n-1)}}{2.0 \cdot (\# \text{ de no convergencias seguidas})} \quad (3.3)$$

Se tiene en cuenta el número de no convergencias seguidas para que, si falla la convergencia de forma reiterativa, se aproxime de forma más rápida al valor adecuado para Δt .

Criterios para actualizar el paso de tiempo (Δt)

La actualización del paso de tiempo es un punto clave en el algoritmo. Además de reducir el paso de tiempo cuando no converge, cuando la iteración converge se debe escoger un valor óptimo del paso de tiempo, es decir, puede ser necesario incrementar o decrementar el paso de tiempo para que la siguiente iteración tenga posibilidades de converger.

- 1) Para calcular el nuevo paso de tiempo se deben comprobar las restricciones generales, que se encuentran resumidas en la ecuación 3.6.

$$\Delta t_{Gen} = \min (\Delta t_{Max}, \Delta t_{Inc}, \Delta t_{Mult}) \quad (3.4)$$

- 2) Para calcular el nuevo paso de tiempo se deben comprobar las restricciones activas referentes a la simulación, que se encuentran resumidas en la ecuación 3.5.

$$\Delta t_{Sim} = \min (\Delta t_{DiffTer}, \Delta t_{Ace}, \Delta t_{Pair}, \Delta t_{V_r}, \Delta t_{Cou}, \Delta t_{Vars}) \quad (3.5)$$

- 3) Para calcular el nuevo paso de tiempo se deben comprobar las restricciones activas referentes al control de las variables, que se encuentran resumidas en la ecuación 3.6.

$$\Delta t_{Vars} = \Delta t^{(n-1)} \cdot \min (\Delta t_{\rho}, \Delta t_U, \Delta t_T) \quad (3.6)$$

Máximo paso de tiempo	Δt_{Max} ; El parámetro Δt_{Max} especifica el valor límite que queremos que tome como máximo.
Incremento máximo del paso de tiempo	$\Delta t_{Inc} = \eta_{Inc} \cdot \Delta t^{(n-1)}$; El parámetro η_{Inc} especifica el incremento máximo respecto del paso de tiempo anterior, habitualmente su valor se sitúa en el rango $[1.1 \div 2.0]$
Multiplicador del paso de tiempo	$\Delta t_{Mult} = \eta_{Mult} \cdot \Delta t^{(n-1)}$; El parámetro η_{Mult} especifica un incremento respecto del paso de tiempo anterior y trata de adaptar Δt a la evolución del sistema, su valor varía con la convergencia de las iteraciones (véase Sec. 3.3.3) y se sitúa en el rango $[1.0 \div \eta_{Inc}]$. Si la iteración no converge se decrementa el parámetro, si la iteración converge se incrementa el parámetro

Cuadro 3.1: Restricciones generales al Δt .

Restricciones de la simulación	Ajuste de Δt
Difusión térmica	$\Delta t_{DifTer} = \min \left[\frac{\rho_i \cdot C_{v_i}}{\kappa_i} \cdot h_i^2 \right]_{i=1}^N$ (C_v es el calor específico a volumen constante; κ es la conductividad térmica y se encuentra definidos en el apartado 2.2.6)
Aceleración	$\Delta t_{Ace} = \min \left[\left(\frac{h_i}{\ \mathbf{a}_i\ } \right)^{1/2} \right]_{i=1}^N$
Condición de Courant (1)	$\Delta t_{Cou1} = \eta_C \cdot \min \left[\frac{h_i}{C_{s_i}} \right]_{i=1}^N$ ($\eta_C = [0.1 - 0.4]$)
Condición de Courant (2)	$\Delta t_{Cou2} = \min \left[\frac{h_i}{C_{s_i} + \mathbf{v}_i + 0.6 \left(\alpha \cdot C_{s_i} + \beta \cdot \max [\mu_{ij}]_{j=1}^N \right)} \right]_{i=1}^N$ (α, β, μ_{ij} se encuentran definidos en el apartado de la viscosidad artificial 2.2.5)

Cuadro 3.2: Restricciones de la simulación al Δt . La restricción de Courant se puede tomar según varias definiciones de menor a mayor complejidad.

Resumiendo, para establecer el paso de tiempo tenemos que tener en cuenta una serie de restricciones generales (cuadro 3.1), restricciones de simulación (cuadro 3.2) y restricciones en el control de variables (cuadro 3.3), aplicando el mínimo de todos ellos según se describe en la ecuación 3.7.

$$\Delta t = \min (\Delta t_{Gen}, \Delta t_{Sim}, \Delta t_{Vars}) \quad (3.7)$$

Restricciones en el control de las variables	Ajuste de Δt
Densidad (ρ)	$\Delta t_\rho = \eta_\rho \cdot \min \left(\frac{\rho_i - \rho_i^{(n-1)}}{\rho_i} \right)_{i=1}^N$
Energía (U)	$\Delta t_U = \eta_U \cdot \min \left(\frac{U_i - U_i^{(n-1)}}{U_i} \right)_{i=1}^N$
Temperatura (T)	$\Delta t_T = \eta_T \cdot \min \left(\frac{T_i - T_i^{(n-1)}}{T_i} \right)_{i=1}^N$

Cuadro 3.3: Restricciones en el control de variables al Δt , basado en ratio de cambio de las variables. Los parámetros $\eta_\rho, \eta_U, \eta_T$ especifican los incrementos máximos respecto el valor anterior de la variable, estando los valores típicos en el rango $[0.02 \div 0.04]$, es decir, entre el 2 % y el 4 %.

3.3. Refinamiento

En la parte del código referida a los refinamientos se pueden especificar tres pasos básicos con el objetivo de conseguir la convergencia requerida. En el siguiente esquema se relacionan esos tres pasos básicos con su coste computacional:

1. Se calculan los vecinos de las partículas, las ecuaciones, componentes físicos adicionales y sus derivadas. El coste computacional de estas operaciones escala como $(n \log n)$.
2. Se calculan las correcciones a la estimación inicial, esto es propiamente el refinamiento. En este caso la naturaleza del cálculo implícito conduce a un sistema lineal enorme que tiene que ser resuelto mediante la inversión de una matriz. El coste de invertir una matriz general de tamaño $n \times n$ mediante un método directo escala como $\theta(n^3)$ (eliminación gaussiana, factorización LU, etc.). Sin embargo, el coste computacional de invertir una matriz dispersa escala como un valor variable según $\theta(n^\alpha)$ donde el parámetro α toma valores en el intervalo $[1, 3]$ dependiendo de la dispersión matriz y la compactación dentro de la dispersión, situándose el valor habitual de α igual o mayor a 2.
3. Se actualizan los valores y se comprueba la convergencia en la iteración. Si no converge realiza un nuevo refinamiento del método Newton-Raphson. Por lo general, se obtiene una convergencia aceptable a partir de 3-4 refinamientos por iteración. La convergencia en la iteración se consigue comprobando las ecuaciones, lo cual implica el recalcu de los vecinos (se repite el paso 1) que escala como $(n \log n)$ y marca el coste de este paso 3. Si en el

número de refinamientos máximos no se alcanza la convergencia de la iteración, es decir la precisión requerida para todas las ecuaciones de todas las variables no se consigue, se recuperan los valores de las variables del anterior paso de tiempo estable, el paso de tiempo se reduce y la iteración se repite de nuevo, empezando desde el primer refinamiento. Si la iteración alcanza la convergencia en algún refinamiento, se almacenan los valores de las variables como las nuevas variables estables, se incrementa la iteración estable, se inicializan los refinamientos y se actualiza el paso de tiempo y se empieza con la siguiente iteración.

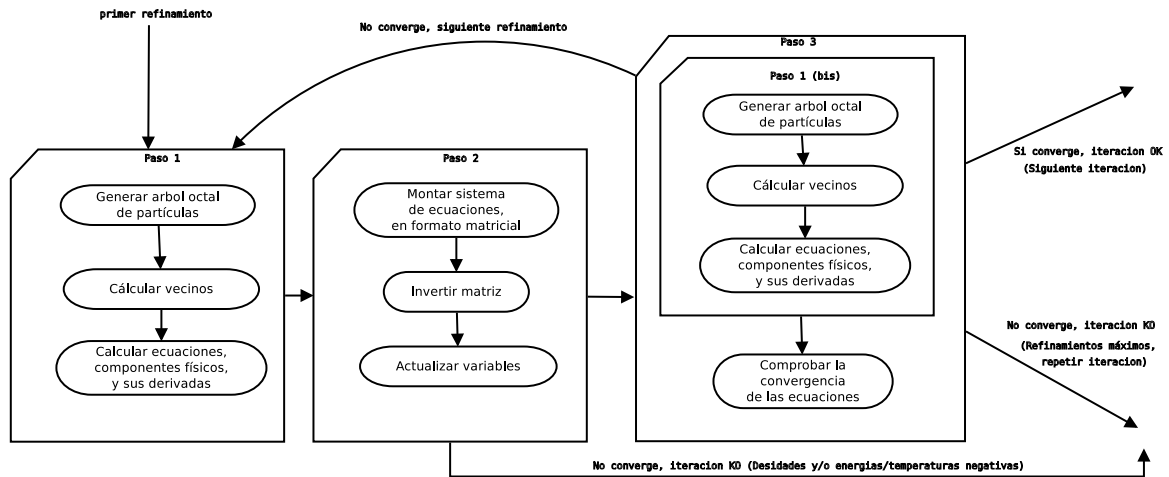


Figura 3.3: ISFAA: Diagrama de flujo de las iteraciones. Aquí se especifican, a grandes rasgos, la actuación del código en el área 2)Refinar.

3.3.1. Calcular ecuaciones y derivadas

En el primer paso, lo primero que se hace en este paso es calcular los vecinos de cada partícula, que luego se tendrán en cuenta a la hora de calcular las ecuaciones.

Esto se puede hacer de varias formas:

- Mediante el cálculo por interacción directa. Donde se comprueba, por cada par de partículas (i, j) si la partícula j se encuentra en el ámbito de influencia del kernel de la partícula i . Este algoritmo es sencillo pero costoso, escalando del orden de $\theta(n^2)$. Si pensamos que para modelos con muchas partículas el coste de invertir la matriz es igual o superior a este, es una buena alternativa para modelos masivos puesto que su coste computacional queda enmascarado y su sencillez de implementación es máxima. Además este coste permite calcular la gravedad de forma exacta por interacción directa, que escala como $\theta(n^2)$.
- Mediante el cálculo en un árbol octal. Es un algoritmo más complejo pero mucho más astuto. Se realizan divisiones sucesivas del dominio espacial en subcubos (en 3D) de modo

que podemos descartar que grupos completos de partículas sean vecinas de otras que se encuentren en subcubos alejados. El coste computacional escala como $\theta(n \log n)$. Utilizar este algoritmo con un coste mas ajustado lleva aparejado que el resto de cálculos, si exceptuamos la inversión de la matriz, se han de ajustar a este coste máximo, por tanto la gravedad se ha de calcular utilizando la aproximación cuadrupolar mediante el potencial gravitatorio, en el árbol octal de búsqueda de vecinos, con un coste de $\theta(n \log n)$

Búsqueda de vecinos de las las partículas mediante Árbol octal (en 3D)

Según se describe en *Hernquist y Katz, 1989* [38], primeramente se genera el árbol de partículas con un coste $\theta(n \log n)$. Para ello se realizan varias acciones: 1) Crear el nodo raíz con una lista de punteros a todas las partículas del modelo. 2) Establecer las distancias máximas del cubo que englobará a todas las partículas. 3) Subdividir el cubo inicial del nodo en 8 subcubos, almacenar sus límites espaciales y crear una nueva lista de partículas en cada subcubo, que será menor o igual a la anterior con solamente las partículas pertenecientes a ese subcubo. 4) Repetir el paso 3 hasta que en todos los subcubos solo haya 1 partícula (figura 3.4).

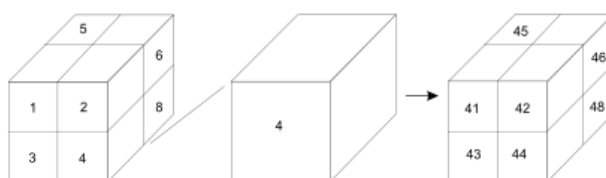


Figura 3.4: Árbol Octal: Distribución del espacial.

Una vez tenemos las partículas distribuidas en el árbol, se buscarán los vecinos de todas las partículas con un coste $\theta(n \log n)$. Para ello, por cada partícula, iremos bajando por el árbol comparando la posición de la partícula más 2 veces la distancia de interpolación, con los límites de los subcubos. De esta forma encontraremos las zonas del dominio donde puede tener vecinos.

Calcular ecuaciones

Una vez obtenidos los vecinos de cada partícula resulta trivial aplicar las formas suavizadas SPH de las derivadas temporales de las variables. Los valores de estas ecuaciones serán introducidas en la matriz para resolver el sistema de ecuaciones.

3.3.2. Calcular las variaciones de las variables

En el segundo paso primeramente se montará el sistema de ecuaciones en forma matriz y se calcularán, mediante un método de Newton-Raphson, las correcciones a cada una de las variables de cada partícula. Con el fin de resolver el sistema de ecuaciones se pueden utilizar dos estrategias: 1) Implementar un método de resolución de sistemas (LU, etc.) o 2) Buscar algoritmos ya implementados para la resolución de matrices que se puedan adaptar a nuestro problema. Estos algoritmos se encuentran muy testeados y minimizan el tiempo de cálculo y el consumo de memoria, buscando un compromiso entre tiempo de cálculo y memoria, ya que minimizar una de las dos condiciones deriva en maximizar la otra.

Montaje del sistema de ecuaciones: Distribución de la matriz

La matriz está compuesta por bloques cuadrados de una cantidad de celdas dependiente de las variables independientes que se escogen, como casos más habituales tenemos (sección 2.3.4): 1) En caso de contemplar variaciones solo en posiciones el número de celdas por bloque es $3 \times 3 = 9$, 2) Si se contempla variación en energía/temperatura el número de celdas por bloque es $4 \times 4 = 16$, 3) Si no agrupamos la velocidad y la densidad en la variación de posiciones el número de celdas por bloque es $8 \times 8 = 64$.

En la matriz hay dos tipos de bloques diferentes, que se encuentran agrupados en filas y columnas contiguas de la matriz. De este modo las celdas contenidas en el bloque son un grupo de ecuaciones de una única partícula, que se derivan respecto de las variables independientes la misma partícula (bloques diagonales) o de otra partícula (bloques de interacción):

- Bloques de la diagonal: Se sitúan en la diagonal y se obtienen cuando las derivadas se calculan respecto de las variables independientes de la partícula actual de la cual estamos tratando sus bloques. Los bloques son la suma de contribuciones de las partículas vecinas.
- Bloques de interacción: Se sitúan fuera de la diagonal y se obtienen cuando las derivadas se calculan con respecto de las variables independientes del resto de partículas diferentes de la que estamos tratando sus bloques. Todas las celdas de bloques de partículas que no sean vecinas de la partícula actual serán cero.

El número de bloques depende del número de vecinos objetivo por partícula, que se establece como una constante al inicio de la simulación. Cuando el número de bloques aumenta la dispersión de la matriz decrece y se vuelve más general.

Si la distribución espacial de los vecinos difiere mucho de la ordenación en la matriz, aumenta la distancia entre los bloques de interacción y también con respecto a la diagonal, esto influye mucho en el coste computacional porque la matriz es menos compacta. Por tanto, en cada iteración o en cada pocas iteraciones es necesario reorganizar los índices para que las partículas que están próximas espacialmente tengan índices próximos en la matriz. Para ellos se pueden utilizar ordenaciones como la ya comentada ordenación de *Morton* [64] que intenta aproximarse a esta distribución (sección 2.3.4)

Bibliotecas de algoritmos de resolución de matrices dispersas

Una parte esencial de la investigación es establecer como se debe realizar la resolución del sistema de ecuaciones y, más concretamente, como realizar la inversión de la matriz. La elección del algoritmo de resolución es clave porque es la parte que tiene mayor coste algorítmico para modelos con muchas partículas. Por nuestra experiencia, esta por encima del 95 % de tiempo de cálculo en sistemas a partir de 20.000 partículas en 3D, aunque es menos importante en sistemas con menor número de partículas. Para el tratamiento es bastante más sencillo utilizar una biblioteca de métodos matemáticos ya implementados para la resolución de sistemas de ecuaciones que implementarlo uno mismo.

Para poder ejecutar cualquier algoritmo es necesario agrupar las celdas no vacías de la matriz en una estructura, tratable por las bibliotecas de algoritmos, de tal modo que sea lo más dispersa posible e intentando que siga un patrón definido. Una vez comprobada la forma de la matriz compararemos esa forma con una serie de matrices estándar conocidas y catalogadas. Según apreciamos en artículos comparativos de algoritmos, como *Gould y colaboradores, 2007* [30], se evalúan las diferentes bibliotecas de algoritmos en función de unas matrices estándar obtenidas de catálogos, derivadas de problemas conocidos. La idea era comparar la forma de esas matrices con la del SPH implícito y escoger la más parecida, con el objetivo de comprobar que bibliotecas algorítmicas podrían funcionar mejor para nuestro problema.

Además, se necesita un método para resolver matrices no simétricas. El SPH variacional utiliza por cada par de partículas $i - j$, los parámetros h_i y h_j . Es decir como no usa la h simetrizada, ni se simetriza el kernel (excepto para gravedad, viscosidad y conducción térmica), la influencia que ejerce i sobre j puede diferir respecto de la que hace j sobre i . Además, en caso de usar h simetrizada o kernel simetrizado, solo sería una matriz simétrica por bloques de ecuaciones, es decir, el bloque b_{ij} sería igual al bloque b_{ji} , pero no tiene porque ocurrir que la celda de la matriz A_{ij} , sea igual al elemento A_{ji}

Como indica *Tournier y colaboradores* [99] los algoritmos de resolución de sistemas, dependiendo de la técnica que implementen, se distribuyen según lo que indica la figura 3.5.

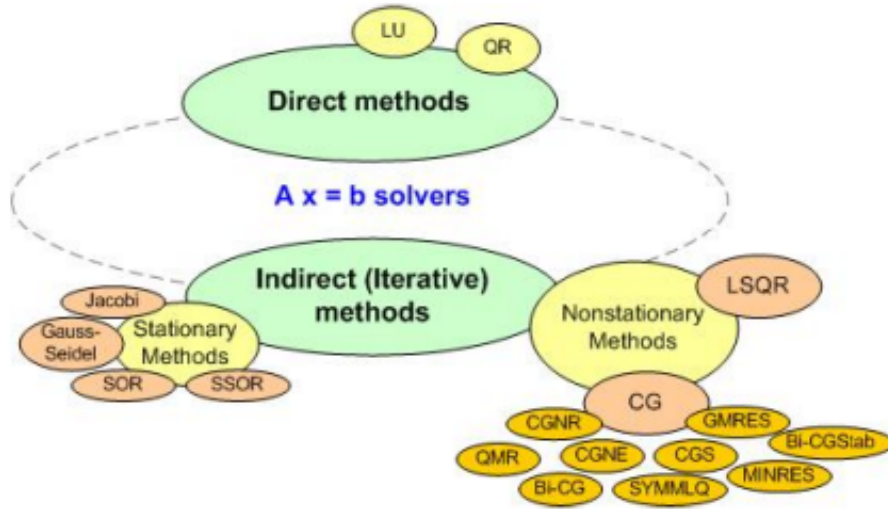


Figura 3.5: Tipos de métodos para resolver sistemas de ecuaciones lineales.

Cuando se comenzó esta tesis se decidió que existían tres candidatos válidos para la implementación: PARDISO, WSMP y MUMPS; Aunque también se valoraron otros métodos como: ScaLAPACK, UMFPACK, etc. pero al final fueron desestimados en favor de los anteriores, porque los anteriores son más rápidos en la convergencia con una exactitud de resultados similar. Como PARDISO era libre para su uso académico y utilizaba métodos directos de cálculo, fue la escogida frente a WSMP propiedad de IBM y de pago y MUMPS que proporcionaba un menor rendimiento que las anteriores. Con el paso del tiempo, WSPM esta ha ido cayendo en desuso pero su autor a recuperado el control del proyecto ofreciendolo libre para entornos académicos y PARDISO se ha vuelto de pago, aunque ha sido incluida en las librerías Intel MKL que se distribuyen par uso libre en su versión de 2016.

Hoy en día, el código ISFAA 4.0 implementa PARDISO 5.0 a la espera de migrar a la versión de PARDISO que se implementa en Intel MKL 2016. Como trabajo futuro estará implementar métodos de resolución de sistemas de ecuaciones sobre GPUs (utilizando librerías como cusp y cuSPARSE) y ejecutar sobre arquitecturas Intel Xeon Phi, que seguro que darán un salto cualitativo muy importante al código.

Todas las bibliotecas dedicadas a resolver sistemas lineales han sido diseñadas para aprovechar la dispersión de la matriz, con lo cual las celdas cero no se almacenan en las estructuras de memoria. Esta propiedad es clave, ya que el factor de ocupación de la matriz típico de nuestros escenarios astrofísicos habituales es $\leq 2\%$. La mayor parte de las bibliotecas utilizan almacenamiento de la matriz comprimido por filas (“compressed by rows” o CSR).

Compresión de la matriz: Formato CSR Para reducir el uso de memoria, la mayoría de bibliotecas, solicitan que se les pase la matriz dispersa en el formato estándar CSR, que solo almacena las casillas no cero de la matriz. Este formato contiene tres vectores: 1) *ia*: Contiene el número de elementos no cero de cada fila y tiene un tamaño de $n+1$, siendo n el número de filas. 2) *ja*: Contiene los índices de las columnas de las celdas no cero de la matriz. 3) *a*: Contiene el valor de cada celda no cero de la matriz. El primer vector es el único de tamaño fijo y sirve para saber la composición de los elementos de los vectores 2 y 3.

Con esta configuración podemos almacenar el valor de cada uno de los elementos no cero de la matriz dispersa. En particular, los índices de columna de la fila i se almacenan a partir de $ja(ia(i))$ y terminan en $ja(ia(i) - 1)$, es decir, el número de celdas no cero de la fila i de la matriz es $ia(i + 1) - ia(i)$.

Podemos observar un ejemplo de almacenamiento CSR en la figura 3.6.

	0	1	2	3	4
0	2.0		3.5		6.7
1		8.2		9.2	
2		1.1	2.8		
3	3.0		1.5	4.5	
4		2.5		8.9	

	0	1	2	3	4	5			
ia	0	3	5	7	10	12			

	0	1	2	3	4	5	6	7	8	9	10	11
ja	0	2	4	1	3	1	2	0	2	3	1	3

	0	1	2	3	4	5	6	7	8	9	10	11
a	2.0	3.5	6.7	8.2	9.2	1.1	2.8	3.0	1.5	4.5	2.5	8.9

Figura 3.6: Matriz dispersa: Ejemplo de compresión CSR.

PARDISO Después de probar algunas bibliotecas de algoritmos paralelos para resolver sistemas de ecuaciones, seleccionamos PARallel Direct SOLver of Basel University, PARDISO V 5.0.0 (*Schenk y Gärtner, 2004* [84]; *Schenk y Gärtner, 2006* [85]; *Schenk y colaboradores, 2007* [86]; *Schenk y colaboradores, 2008* [83]) por su eficiencia, su licencia gratuita para entornos académicos de investigación (con validez de 1 año, renovable), pero de pago para entornos comerciales. Además Intel la ha incluido en sus librerías MKL 11.3 de libre uso y distribución en su versión de 2016.

Es un algoritmo de alto rendimiento, con tratamiento robusto de memoria, eficiente y fácil de usar, para resolver grandes sistemas lineales dispersos simétricos y asimétricos de ecuaciones en multiprocesadores de memoria compartida o distribuida.

Puede tratar sistemas asimétricos, estructuralmente simétricos, completamente simétricos,

reales o complejos, positivos definidos o indefinidos y hermitianos. Para ello utiliza LU con pivoteo completo mediante multiprocesadores con OpenMP y multicomputadores con paso de mensajes MPI. Utiliza una combinación automática de algoritmos iterativos y directos para acelerar el proceso de solución en los grandes sistemas de ecuaciones.

Se puede conseguir el código, la documentación y la información de las licencias en la siguiente web de los autores (visitada por última vez el 09/11/2015): <http://www.pardiso-project.org>.

WSMP La Watson Sparse Matrix Package (WSMP) es una colección de algoritmos para resolver eficientemente grandes sistemas de ecuaciones, cuyos coeficientes derivan en una matriz dispersa. Tiene alto rendimiento, es robusta, su software es fácil de usar y se puede utilizar en serie, en paralelo con un multiprocesador con memoria compartida o escalable con paso de mensajes en multicomputadores con memoria distribuida. Ha sido desarrollado por Anshul Gupta para IBM (*Gupta, 2000* [33], *Gupta, 2002* [34], *Gupta, 2007* [32]) y *Gupta y colaboradores, 2009* [35]) y aunque inicialmente fue de pago, hoy en día, es libre para entornos académicos y se pueden conseguir licencias que se van renovando por periodos de entre 6 meses a 1 año.

Actualización de las variables

Una vez que el algoritmo de resolución del sistema ha terminado se obtienen en un vector las correcciones (que denominaremos δ) de todas las variables independientes de todas las partículas.

Actualización de posiciones y energías/temperaturas. Estas correcciones se aplican sobre las variables con las ecuaciones del cuadro 3.4 y se pasa al siguiente paso, donde se comprobará la corrección de las nuevas variables con las ecuaciones.

$x = x^{n-1} + \delta x \quad ; \quad y = y^{n-1} + \delta y \quad ; \quad z = z^{n-1} + \delta z$
$U = U^{n-1} + \delta U \quad o \quad T = T^{n-1} + \delta T$

Cuadro 3.4: Variables independientes: Posiciones y energía/temperatura

Actualización de velocidades y densidades. La actualización de las velocidades y densidades se puede realizar de dos formas: 1) Como variables independientes del sistema, o 2) Si se

han compactado con las posiciones existen dos métodos: Calculando directamente las ecuaciones ó aplicando las correcciones de las posiciones. Estos métodos se describen en el cuadro 3.5.

$v_x = v_x^{n-1} + \delta v_x$	$v_x(x, x^{n-1}) = \frac{x - x^{n-1}}{\Delta t} \implies \delta v_x = \frac{\left(\frac{x - x^{n-1}}{\Delta t}\right)}{\partial x} \cdot \delta x = \frac{1}{\Delta t} \cdot \delta x$
$v_y = v_y^{n-1} + \delta v_y$	$v_y(y, y^{n-1}) = \frac{y - y^{n-1}}{\Delta t} \implies \delta v_y = \frac{\left(\frac{y - y^{n-1}}{\Delta t}\right)}{\partial y} \cdot \delta y = \frac{1}{\Delta t} \cdot \delta y$
$v_z = v_z^{n-1} + \delta v_z$	$v_z(z, z^{n-1}) = \frac{z - z^{n-1}}{\Delta t} \implies \delta v_z = \frac{\left(\frac{z - z^{n-1}}{\Delta t}\right)}{\partial z} \cdot \delta z = \frac{1}{\Delta t} \cdot \delta z$
$\rho = \rho^{n-1} + \delta \rho$	$\rho(x, y, z) = \sum_{j=1}^N m_j W_{ij}(h_i) \implies \delta \rho = \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial x} \cdot \delta x$ $+ \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial y} \cdot \delta y$ $+ \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial z} \cdot \delta z$ $+ \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h} \cdot \delta h$

Cuadro 3.5: Variables independientes o dependientes: Velocidades y densidad.

Aviso de negativos en energías/temperaturas o densidades. Cuando se está realizando la actualización de las variables independientes puede ocurrir que la corrección derivada del resultado de la matriz provoque que una o varias de estas variables se haga negativa. En este caso se abortará el refinamiento y la iteración, porque da un resultado que no es físico. Se reiniciará la iteración empezando de nuevo el primer refinamiento y con un paso de tiempo menor, para intentar que no vuelva a aparecer el problema.

Fijar la distancia de interpolación inicial (h_i) Para fijar la distancia de interpolación inicial h del modelo, se utiliza lo que se ha denominado vecinos fraccionarios (eq. 3.8), donde el número de vecinos real se trasforma en fraccionario a través de la distancia al vecino \mathbf{r}_{ij} y la distancia de interpolación h_i .

$$n_{v_i} = \sum_{j=1}^N \left[1 - \left(\frac{\mathbf{r}_{ij}}{2 h_i} \right)^2 \right] \quad (3.8)$$

Mediante la ecuación 3.9 se actualiza el valor de h , intentando que una partícula encuentre el número de vecinos objetivo ($n\Phi$) marcado, cuyo valor suele estar entre 30 y 100 reales que

corresponden a un valor $n\Phi = [12, 25]$ fraccionarios en los test realizados.

$$h_i = h_i^{(n-1)} \cdot \frac{1}{2} \left(2^{h_\nu} \frac{n\Phi}{\max(10, nv_i)} \right)^{1/h_\nu} \quad (3.9)$$

El parámetro h_ν se escoge de tal modo que cuando el número de vecinos ideal coincide con el real la h no varía, siendo su rango de valores $h_\nu = [2, 10]$ y su valor habitual $h_\nu = 3$. La elección se realiza en función de la suavidad con la que se quiere aproximar la h , siendo los valores bajos aproximaciones rápidas y los altos aproximaciones lentas, a mayor valor mejor aproximación pero más iteraciones necesita para llegar al valor correcto.

Durante la ejecución del código implícito la h se calcula como una variable independiente más, ya que esta integrada en el formalismo Lagrangiano.

3.3.3. Comprobar la convergencia

En el tercer paso, una vez se han actualizado las partículas, se necesita repetir el paso 1 para conseguir los nuevos valores de las ecuaciones de Newton-Raphson a partir de los cuales se comprobar la convergencia de las mismas.

Para conseguir este objetivo se han definido varios sistemas dependiendo de sobre que parte del control de ecuaciones queremos influir. Tenemos dos grupos de restricciones: completas e individuales. Las restricciones completas hacen una selección previa, delimitando sobre que partículas se ha de comprobar la convergencia. Las restricciones individuales afectarán a una o varias de las ecuaciones de las partículas que hayan sido seleccionadas por el filtro completo.

Restricciones Completas: Control previo en las partículas

Una forma de no realizar controles innecesarios es realizar el control solo sobre las partículas de interés para el modelo físico tratado, en una iteración concreta. Por ejemplo, en algunas situaciones puede usarse la ecuación 3.10, de tal modo que solo se comprueban las partículas que cumplan el criterio, es decir, que tengan una cierta velocidad respecto de la velocidad del sonido local. Suponemos que las partículas que no cumplan el criterio no influirán lo suficiente en el modelo y no afectarán a los cambios del paso de tiempo.

$$\frac{|v_i|}{C_{s_i}} \geq 10^{-4} \quad (3.10)$$

Restricciones particulares: variables y ecuaciones

Una vez seleccionadas las partículas que se controlarán se deberán aplicar los controles sobre dos vertientes: 1) Las variables seleccionadas en los parámetros iniciales del programa, 2) Las ecuaciones seleccionadas en los parámetros iniciales del programa.

Control de las variables El control sobre variables afecta a la variación de una variable cualquiera (χ_i) respecto del paso de tiempo anterior ($n-1$) según se describe en la ecuación 3.11. Se puede especificar valores de variación permitido diferentes para cada variable independiente (posición, energía/temperatura, velocidad y/o densidad), siendo 10^{-4} el valor habitual.

$$\frac{|\chi - \chi^{(n-1)}|}{\max(|\chi|, |\chi^{(n-1)}|)} \leq 10^{-4} \quad (3.11)$$

Control de las ecuaciones El control sobre las ecuaciones afecta a la convergencia del método Newton-Raphson y aplica exactamente lo descrito en la sección 2.3.4 de la explicación del método SPH. Según los parámetros iniciales de ISFAA se comprobarán todas las ecuaciones o parte de ellas. Se pueden encontrar todas las ecuaciones implementadas en el apéndice E.

3.4. Coste computacional

El coste computacional de los modelos, con un número de partículas alto, se encuentra determinado por la inversión de la matriz. El coste asociado es consecuencia directa del número de celdas no nulas de la matriz, es decir, su tasa de relleno, dispersión y compactación, Tres son los parámetros que influyen con mayor intensidad en el número de celdas no nulas:

1. El número de partículas: Este parámetro proviene del modelo inicial, por tanto en el estado actual del método los modelos implícitos habitualmente tendrán pocas partículas.
2. El número de vecinos objetivo: El número habitual por partícula es entre $[30, 60]$, a mayor número de vecinos la densidad de la matriz dispersa aumentará y por tanto el esfuerzo computacional del algoritmo se incrementará, aumentando α .
3. El número de variables independientes por partícula: Esto se decide en función del modelo.

3.5. Funcionalidades adicionales

Además de todo lo descrito anteriormente se han implementado una serie de complementos necesarios para simular algunos modelos concretos.

- *Aproximar variables (refinamiento 0)*: Se tiene en cuenta las dos iteraciones estables anteriores para aproximar las posiciones a su valor real en el paso de tiempo que se está probando y así iniciar el refinamiento a partir de un valor de las posiciones probablemente más cercano al valor real buscado. El cambio de posición implica el recalcu de la densidad.
- *Pairing-instability*: Dado un par de partículas, cuando la distancia entre ellas es muy pequeña se pueden quedar “pegadas”, y esto paraliza la simulación y provoca que para conseguir la convergencia se disminuya mucho el paso de tiempo. El ajuste de los parametros de los kernels Sinc eliminan o reducen mucho el impacto de este problema.

Capítulo 4

Tests básicos

La validación del código ISFAA se realiza a través del análisis de varios test con resultados analíticos conocidos, comprobando que la evolución del sistema es la correcta y que se produce el efecto físico esperado.

En cada caso de test se sigue el siguiente esquema:

- Descripción: Donde se analiza lo que se va a probar y lo que se espera obtener.
- Modelo inicial: Donde se da una descripción física del modelo, con los valores necesarios en las variables y los ingredientes físicos que intervienen en el modelo.
- Resultados: Donde se analizan las salidas de la simulación.

Por la naturaleza del método implícito, para evitar que el tiempo de cálculo sea excesivo, se han utilizado modelos con un número bajo de partículas (entre 5 000 y 20 000 partículas dependiendo de los test). Debido a ello, y por su naturaleza lagrangiana, solo las regiones de densidades elevadas se encuentra bien resueltas. Por tanto, para obtener la suficiente resolución la mayoría de los test se muestran en 2D, ya que para obtener resultados precisos en 3D sería necesario al menos unas 100 000 partículas.

Los test que se muestran son los siguientes:

1. Explosión puntual (test de Sedov).
2. The wall heating shock (test de Noh).

3. Crecimiento de inestabilidades de Rayleigh-Taylor¹.
4. Caída libre (Free-Fall collapse).
5. Expansión de una onda térmica.

4.1. Explosión puntual

En el test de Sedov se simula la evolución de una onda de choque que nace como consecuencia de una explosión puntual central. El objetivo es analizar como se propaga en un medio homogéneo la onda de choque producida. La solución analítica fue descrita por *Sedov, 1959* [89] aplicando métodos auto similares y análisis dimensional para diferentes geometrías.

La formulación más simple del test de Sedov tiene un gas ideal frío inicialmente en reposo. En $t = 0$ s. hay una explosión en el punto de origen que se propaga hasta afectar a todo el modelo. Se produce una explosión muy intensa que supone un trasvase continuo entre energía cinética y energía interna. Uno de los objetivos del test es comprobar como se comporta la viscosidad artificial, que se encarga de ensanchar el choque en 2 o 3 veces la longitud de suavizado inicial. Para calcular la presión se utiliza la ecuación de estado ideal, con un coeficiente adiabático de $\gamma = 5/3$. La solución analítica predice que en el frente de onda se tiene que observar un salto en densidad de un factor 4.

Para construir la explosión se libera una gran cantidad de energía en una pequeña región central. Esta región tendrá una energía 10.000 veces superior en su parte central, que en su parte exterior. Para suavizar la discontinuidad inicial se hace una distribución en campana de *Gauss* de la energía, cuyo centro es el centro de masas del sistema, de tal modo que la distribución de presión siga la ecuación siguiente:

$$P(r) = P_2 + (P_1 - P_2) \cdot \exp\left(\frac{-r^2}{\sigma^2}\right) \quad (4.1)$$

donde P_1 y P_2 son las presiones en las zonas interna y externa, r es la distancia al centro de masas del sistema, y el parámetro σ establece el ancho de la zona donde la presión decae (en el ejemplo $\sigma = 0.05$).

En más de una dimensión la resolución es crucial, no solo para resolver el frente de onda, sino también para reproducir la zona post-choque y la estructura enrarecida que deja tras de sí en el medio atravesado. El test se ha reproducido en 2D y 3D, pero para observar los resultados con mayor resolución, los datos y las figuras que aquí se muestran pertenecen al modelo 2D.

¹Se comprueban durante la fase final de la compresión de una microcápsula donde se simula el proceso de fusión por confinamiento inercial (*ICF*).

Modelo inicial

Se dispone de una distribución equidistante de 7 668 partículas en 2D con forma circular y con un diámetro de 1 *cm*. Las partículas inicialmente se encuentra en reposo, es decir, al inicio de la simulación ($t = 0$ s) la velocidad de todas las partículas es $v = 0$ *cm/s*, de tal modo que el sistema inicial se encuentra en equilibrio. El valor de la presión es $P_1 = 10^4$ dyn.cm⁻², $P_2 = 1$ dyn.cm⁻² y la anchura de la zona de suavizado de presión es $\sigma^2 = 0.0025$ cm², lo que implica que la variación de presión se distribuye en una región radial equivalente a unas 5-6 veces la longitud de suavizado desde el centro de masas.

La física aplicada en el modelo es: Ecuación del momento en 2D, ecuación de la energía, ecuación de densidad-h, ecuación de estado de gas ideal y viscosidad artificial.

Para probar resoluciones mayores, se ha realizado otra simulación con una distribución de 46 352 partículas en 2D y los mismos parámetros iniciales.

Resultados

El cuadro 4.1 describe el comportamiento del código en diferentes PCs y para diferentes valores de la viscosidad.

Se observa que el Speedup es mucho menor del esperado, cuando la resolución es baja, esto es debido a que una parte del programa es secuencial. La parte paralela esta focalizada en el uso masivo de partículas, donde el tiempo de cálculo aumenta de forma entre cuadrática y cúbica.

Con una resolución más alta ($\simeq 46\,000$ partículas en 2D), el cálculo se centra principalmente en la inversión de la matriz. Con cuatro variables independientes x , y , U y h , la matriz aproximadamente contiene una tasa de relleno del 0.04 %, es decir $15 \cdot 10^6$ celdas no cero de las $4 \cdot 10^{10}$ posibles). En la máquina B, el uso de memoria es de 22 *GBytes* y el tiempo de usuario de la inversión de la matriz se sitúa sobre 100 s, por los 5 s del modelo con pocas partículas. El Speedup aumenta a los valores esperados porque básicamente depende de este cálculo, como ya se ha comentado en capítulos anteriores. Cuando hay pocas partículas la parte secuencial consume el 40 % del tiempo por un 60 % de la parte paralela, sin embargo en el modelo con seis veces más partículas la parte secuencial supone el 10 % del tiempo por un 90 % de la parte paralela. Si aumentamos el número de partículas la diferencia cada vez será mayor.

Con este test se demuestra que en función del número de partículas el coste en memoria es aproximadamente lineal $\theta(n)$, pero el coste computacional es entre cuadrático y cúbico $\theta(n^{[2 \div 3]})$, en función del algoritmo utilizado y la distribución de la matriz de entrada (es decir

Parts.	α	β	PC	$\tau_{Model} (s)$	$\tau_{CPU_{system}} (h)$	$\tau_{CPU_{user}} (h)$	Speedup	Iters.	Refs.
7 887	0.5	1.0	A	0.09186	1.41	0.48	2.375	34	212
7 887	1.0	2.0	A	0.08793	1.59	0.54	2.944	34	209
7 887	1.0	2.0	B	0.08793	1.74	0.36	4.833	34	209
46 352	1.0	2.0	B	0.08343	134.42	12.97	10.364	101	705

Cuadro 4.1: Tiempo de computación del código implícito, para diferentes PCs y en función del comportamiento de la viscosidad artificial (según el valor de los parámetros α y β). Los valores son obtenidos cuando la onda de choque llega a $r = 0.3$ cm desde el centro de masas. La ejecución en el PC A se realiza con 8 threads en un portatil con CPU Intel Core i7-2720QM 2.2Ghz y 16 GB DDR3-1600Mhz. La ejecución en el PC B se realiza con 16 threads en un servidor multinúcleo con 2 CPU Intel Xeon W5580 3.2Ghz y 48 GB DDR3-1333Mhz.

su preconditionamiento).

En la figura 4.1 se muestran los tiempos de computación en los PCs(A,B), para el modelo de baja resolución y las dos pruebas de viscosidad.

En la figura 4.2 se muestran los tiempos de computación para el modelo con más resolución.

En la figura 4.3 se muestran los perfiles de densidad y velocidad de forma simultánea, cuando ya se ha formado completamente el frente de onda a $r = 0.3$ cm del centro de masas y $t \simeq 9$ ms. En el código implícito el pico de densidad es de aproximadamente el 95 % de lo que se espera en un movimiento violento de un gas con $\gamma = 5/3$, con una zona post-choque mucho más regular que en el código explícito estándar. Salvo esta pequeña diferencia, se puede concluir que los dos códigos reproducen los resultados de la solución analítica.

En la figura 4.4 se puede observar la evolución temporal de las partículas del modelo, y en la figura 4.5 la conservación de la energía y el momento lineal del sistema.

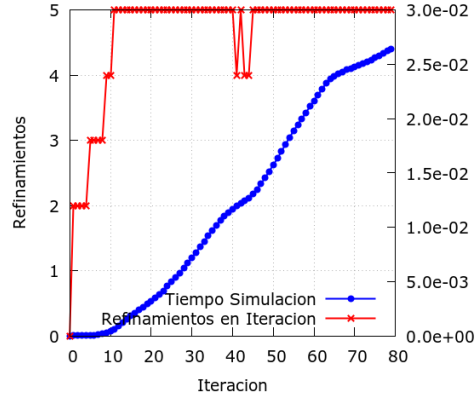
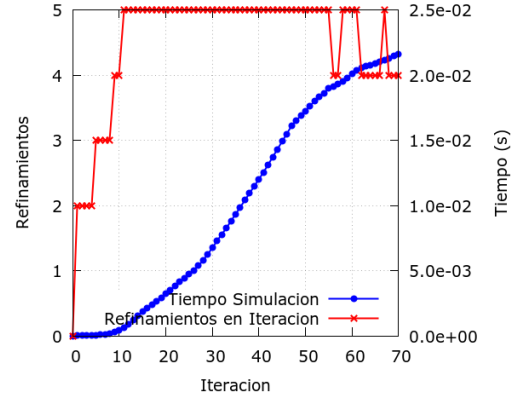
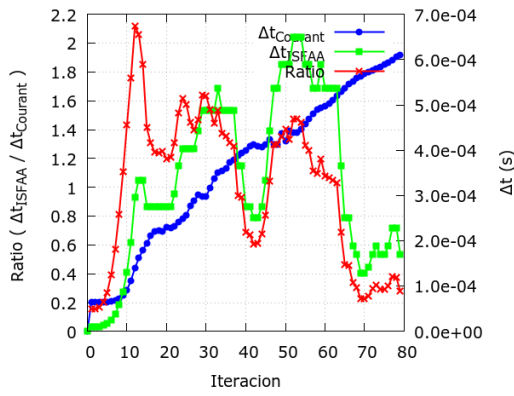
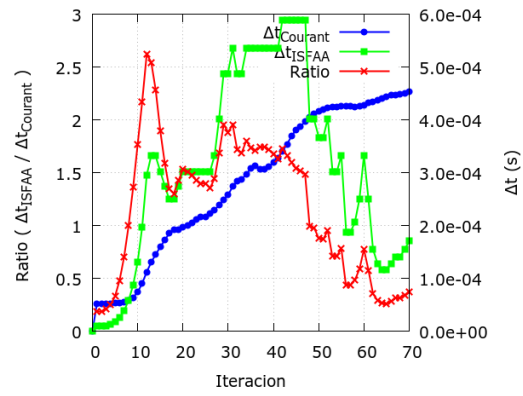
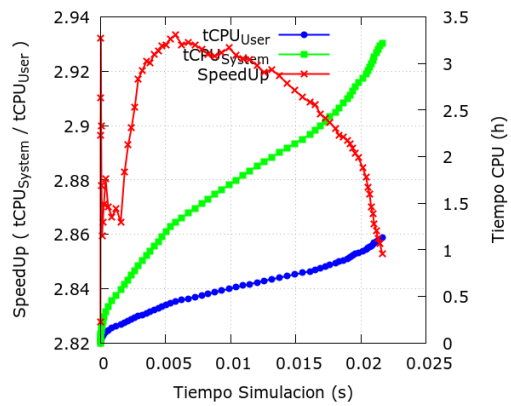
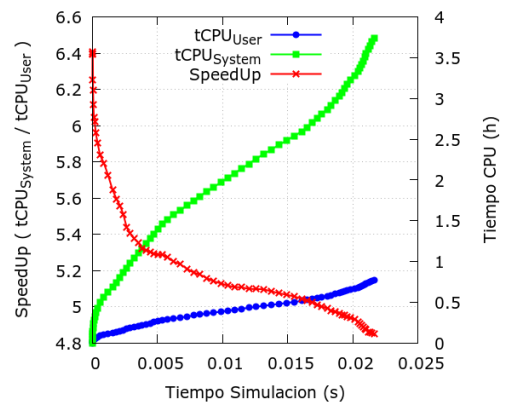
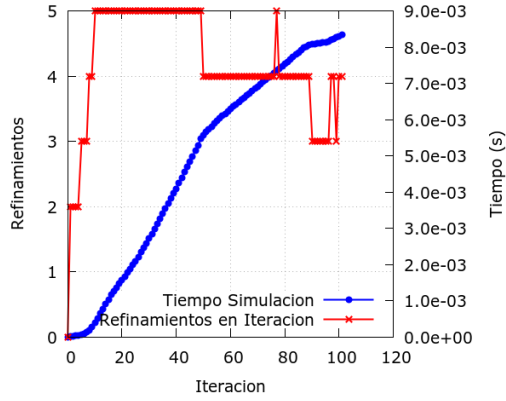
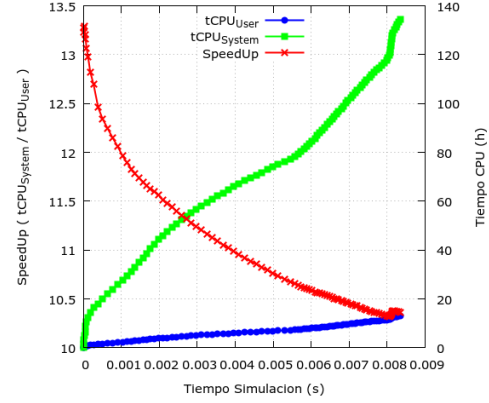
(a) $\alpha = 0.5$, $\beta = 1.0$ - PC(A)(b) $\alpha = 1.0$, $\beta = 2.0$ - PC(A)(c) $\alpha = 0.5$, $\beta = 1.0$ - PC(A)(d) $\alpha = 1.0$, $\beta = 2.0$ - PC(A,B)(e) $\alpha = 1.0$, $\beta = 2.0$ - PC(A)(f) $\alpha = 1.0$, $\beta = 2.0$ - PC(B)

Figura 4.1: Tiempos de computación de la simulación con 7 668 partículas.



(a) Iteración Vs Refinamientos



(b) Speedup

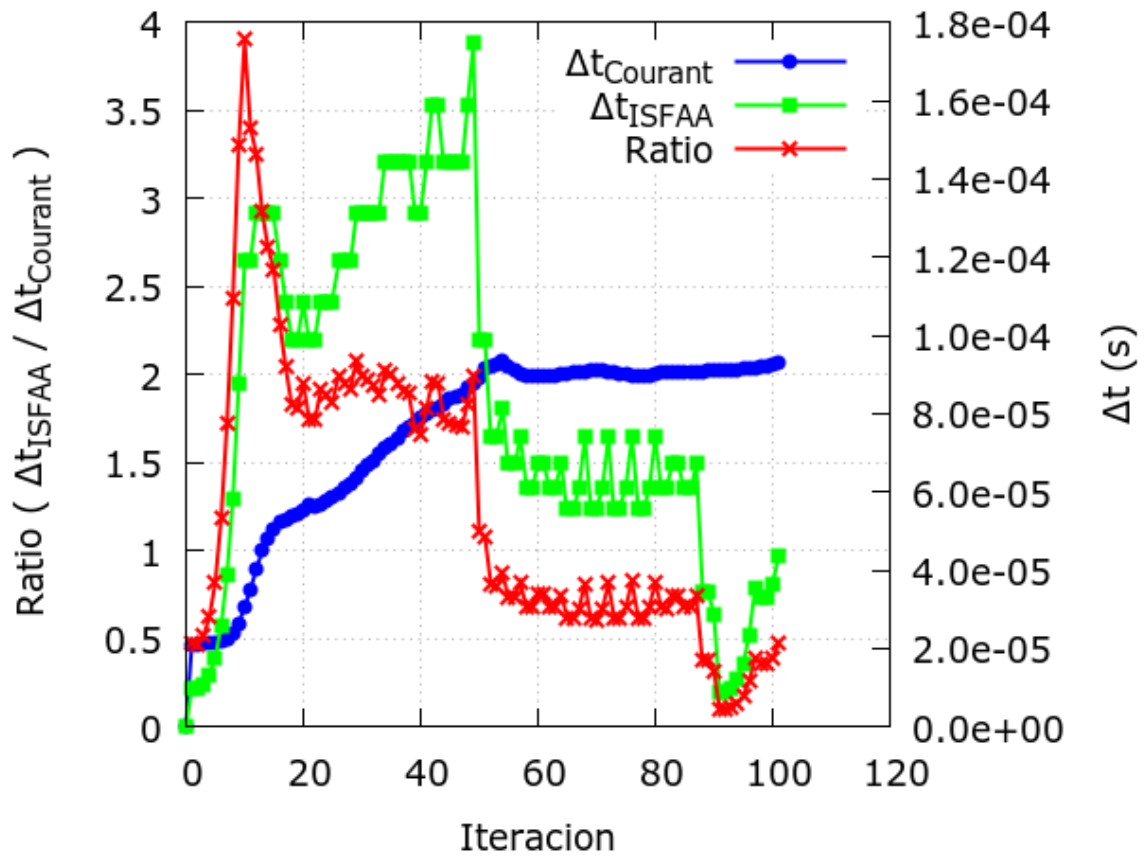
(c) Ratio $\Delta t_{ISFAA}/\Delta t_{Courant}$

Figura 4.2: Tiempos de computación de la simulación con 46 352 partículas.

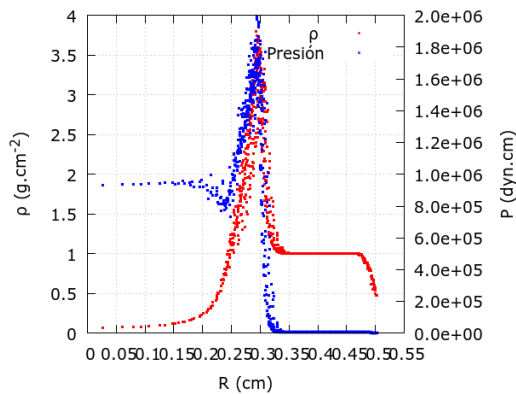
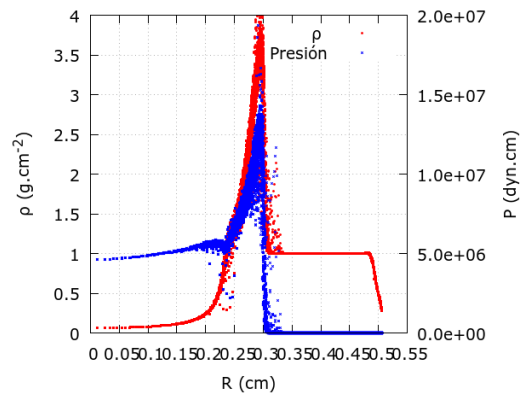
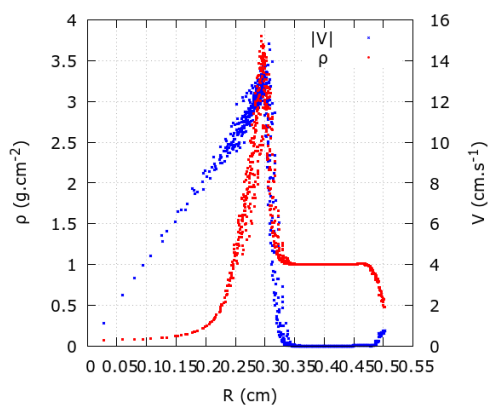
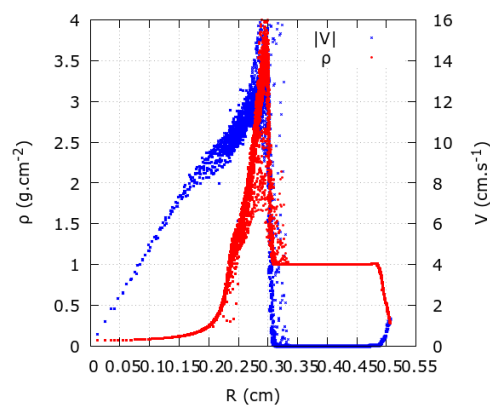
(a) 7 688 Particles (ρ , P)(b) 46 352 Particles (ρ , P)(c) 7 688 Particles (ρ , V)(d) 46 352 Particles (ρ , V)

Figura 4.3: Perfiles de densidad, presión y velocidad en dos modelos con diferente número de partículas a tiempo $t \simeq 9$ ms. Los perfiles de densidad han sido normalizados al valor de la densidad antes del choque. Se puede observar que en el modelo con pocas partículas hay un falta de densidad y sin embargo en el modelo con muchas partículas hay un exceso de densidad, respecto de la solución analítica. Además, también vemos diferencias en el comportamiento de la presión, esto es debido a que las partículas tienen la misma masa en los dos modelos y al tener más partículas tenemos más masa y por tanto más densidad y presión. Salvo estas diferencias, en los dos casos la compresión se sitúa entorno al 90 % del valor teórico.

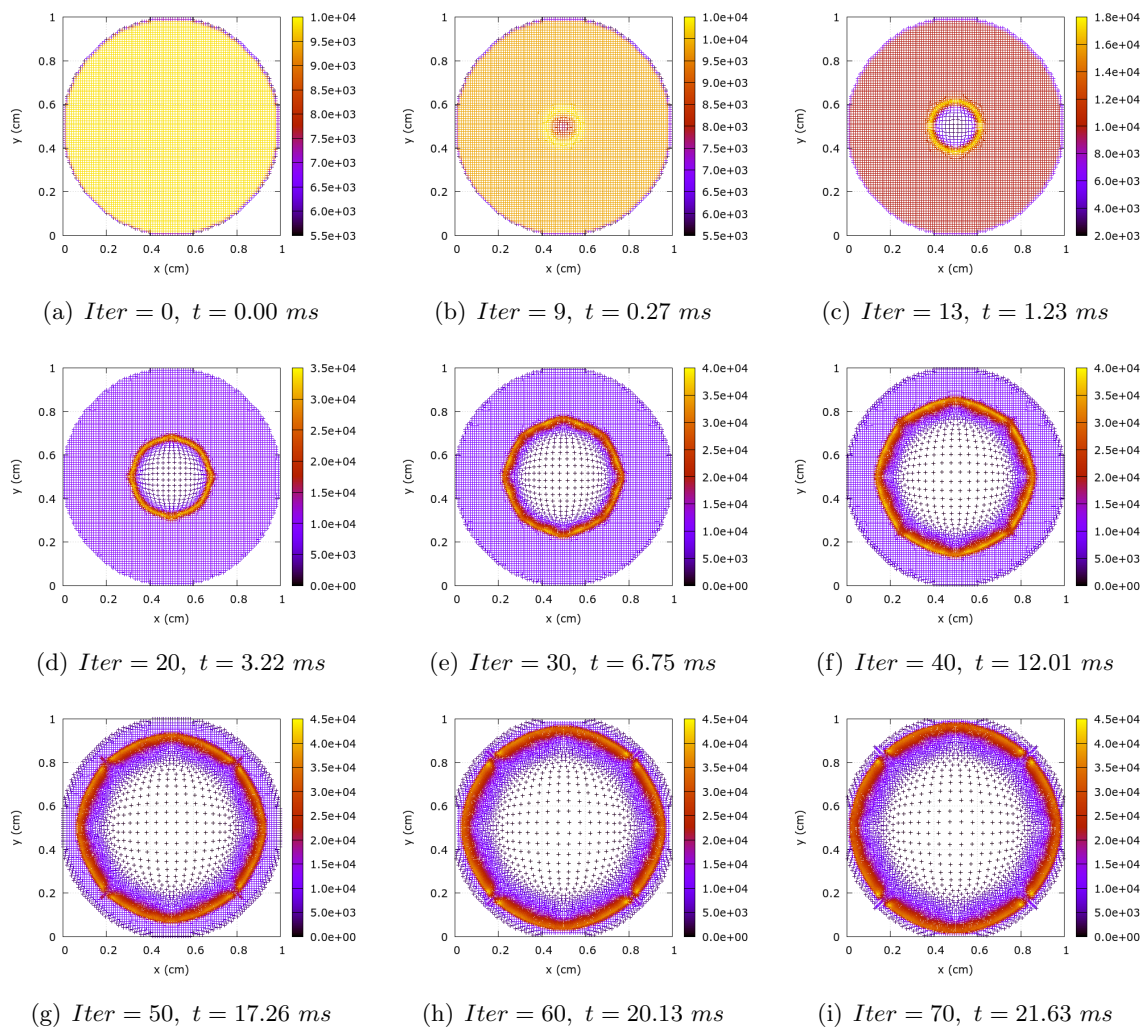
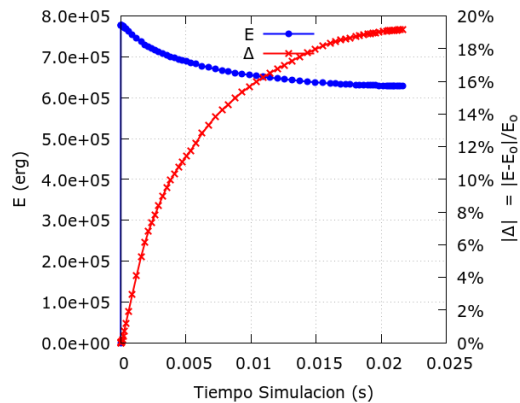
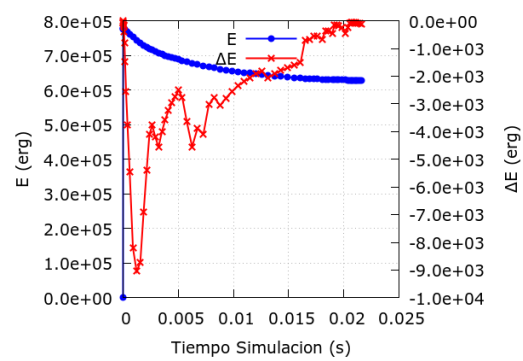


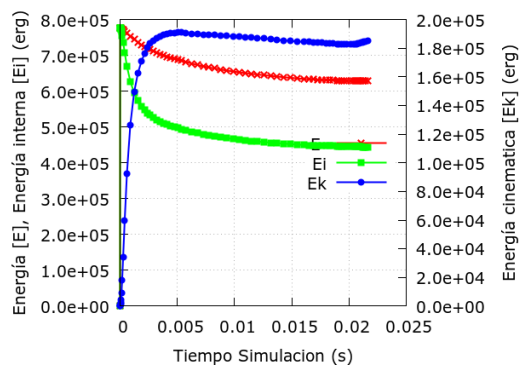
Figura 4.4: Evolución temporal de la simulación con 7 668 partículas y los parámetros de la viscosidad $\alpha = 1$, $\beta = 2$. Se muestra el mapa de color de la densidad (g/cm^2) en diferentes instantes de tiempo.



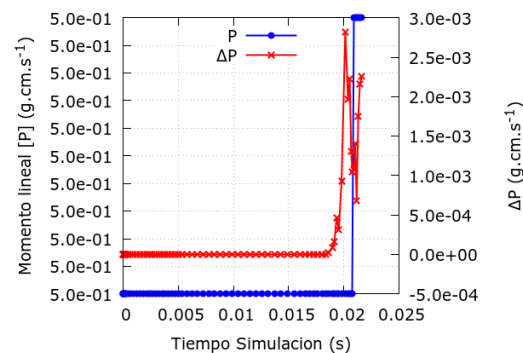
(a) Variación de la energía total del sistema



(b) Conservación de la energía



(c) Energía del sistema



(d) Conservación del momento lineal

Figura 4.5: Conservación de la energía y momento lineal de la simulación con 7 668 partículas. Como se puede apreciar en el panel 4.5(a) existe una pérdida importante de energía de forma inicial, esto es un efecto transitorio mientras se forma la onda. Esto es justificable por la baja resolución del modelo de 7 668 partículas que lo sitúa entorno al 19 %, sin embargo, en el modelo con más resolución de 46 352 partículas esta pérdida se sitúa entorno al 14 %. También se observa en el panel (4.5(d)) la conservación del momento es excelente hasta un tiempo de 0.2 s. En la figura 4.4 se puede ver que dicho momento corresponde a la llegada del frente de choque a la frontera externa del juego de partículas utilizado en este test.

4.2. The wall heating shock

El test “The wall heating shock” fue descrito junto con su solución analítica en *Noh, 1987* [67]. El modelo a simular contiene un fluido, que sigue la ecuación de estado de gas ideal con un coeficiente adiabático de $\gamma = 5/3$, con un flujo de gas supersónico que es lanzado hacia el centro geométrico. Este flujo genera una región de alta compresión.

Las características generales del escenario se pueden describir en SPH con una resolución suficiente de partículas. A pesar de ello, los esquemas que dependen de la viscosidad artificial tienen dificultades para dar una descripción exactamente coincidente con la solución analítica y los resultados son del orden del 80 % del valor esperado. La razón es que la viscosidad artificial extiende el choque sobre varias partículas, lo que introduce un aumento no-físico de la energía interna antes del choque. En este caso, por convergencia de los flujos, un gran pico artificial de energía interna se observa en la parte central. Como consecuencia, aparece una profunda caída en el perfil de densidad para tratar de mantener el suavizado de la presión.

El análisis de los resultados del test ayuda a validar la consistencia del esquema implícito, comprobando que funcionan correctamente los gradientes de energía/temperatura y la disipación producida por la viscosidad artificial.

El test se ha reproducido en 2D y 3D, pero para observar los resultados con mayor precisión, los datos y las figuras que aquí se muestran pertenecen al modelo 2D.

Modelo inicial

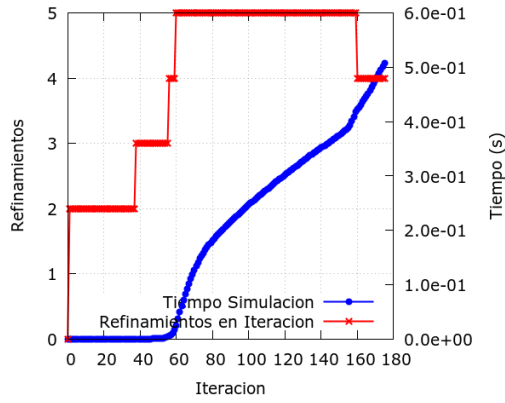
Se dispone de una distribución equidistante de 7 668 partículas en 2D con forma circular y con un diámetro de 1 cm. El valor de la densidad es homogéneo, excepto en las zonas exteriores, y tiene un valor de $\rho = 1 \text{ g.cm}^{-3}$. A la energía interna se le da un valor de $u = 10^{-20} \text{ erg.g}^{-1}$.

Para generar la implosión, se impone un campo de velocidades simétricamente esférico con un valor en la velocidad radial de $v_r = -1 \text{ cm.s}^{-1}$.

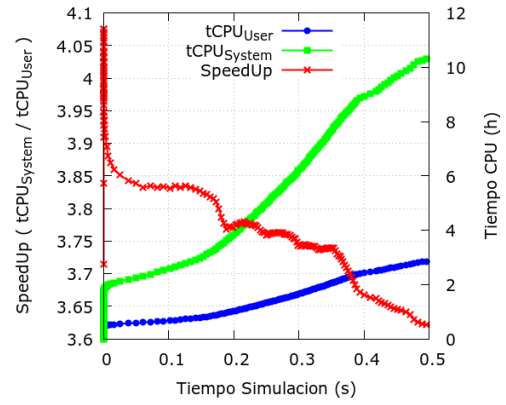
Resultados

El cuadro 4.2 describe el comportamiento de la ejecución del código.

En la figura 4.6 se muestran los tiempos de computación.



(a) Iteración Vs Refinamientos



(b) Speedup

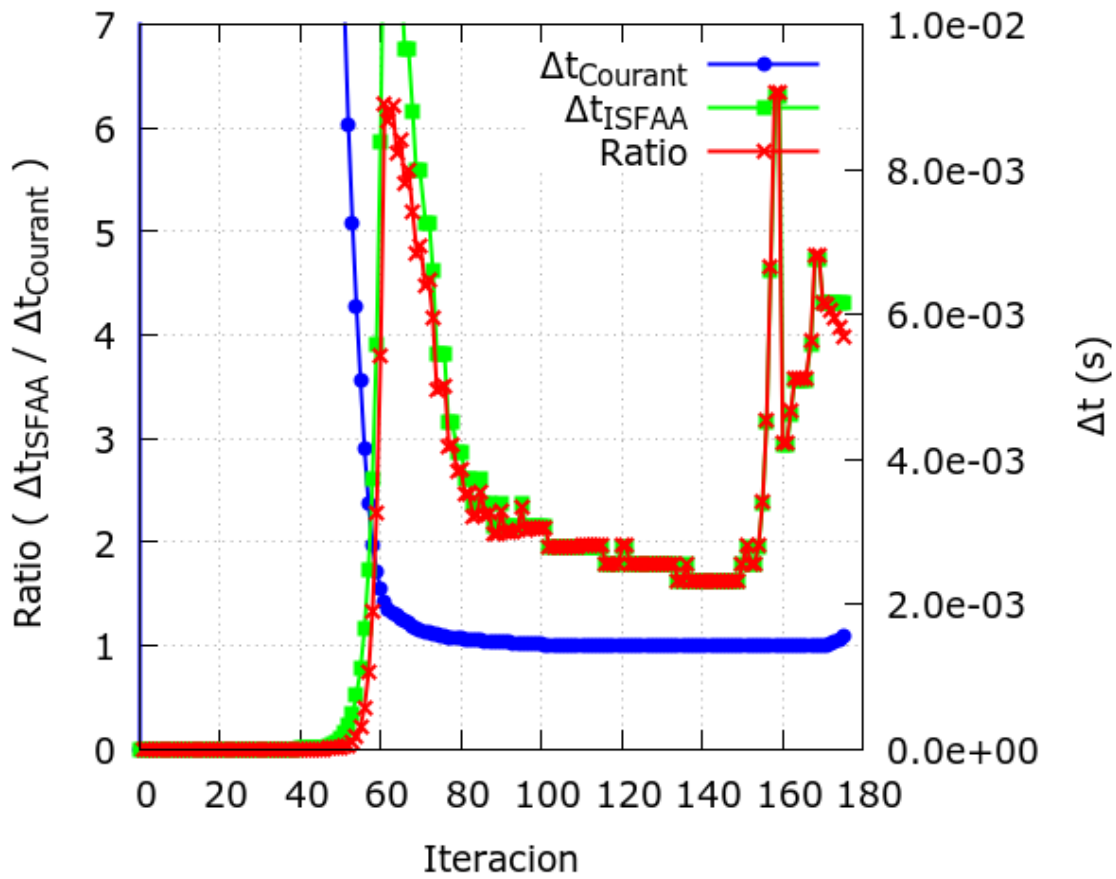
(c) Ratio $\Delta t_{ISFAA} / \Delta t_{Courant}$

Figura 4.6: Tiempos de computación de la simulación con 7 668 partículas.

Parts.	α	β	PC	$\tau_{Model} (s)$	$\tau_{CPU_{system}} (h)$	$\tau_{CPU_{user}} (h)$	Speedup	Iters.	Refs.
7 668	1.0	2.0	A	0.328	6.98	1.86	3.74	130	702

Cuadro 4.2: Tiempo de computación del código implícito en función del comportamiento de la viscosidad artificial. Los valores son obtenidos cuando la onda de choque llega a $r = 0.1$ cm desde el centro de masas. La ejecución en el PC A se realiza con 8 threads en un portatil con CPU Intel Core i7-2720QM 2.2Ghz y 16 GB DDR3-1600Mhz.

En la figura 4.7 se muestran los perfiles de densidad y velocidad radial en el instante $t \simeq 0.3$ s, cuando el choque ya se encuentra totalmente desarrollado y son coincidentes con los cálculos explícitos en *García-Senz y colaboradores, 2009* [27].

En la región central, $r < 0.1$ cm el material se acumula en forma de meseta. Se puede observar que aparece una cierta dispersión por usar pocas partículas y utilizar un esquema implícito puro (a primer orden con $\Theta = 1$). La materia en esta región interior permanece estancada con una velocidad cercana a cero. Fuera de la meseta la densidad disminuye abruptamente a través del frente de choque tratando de recuperar su valor inicial, mientras que se mantiene la velocidad inicial de $v_r = -1$ cm.s⁻¹ en la zona del material que todavía no se ha visto involucrada en el choque.

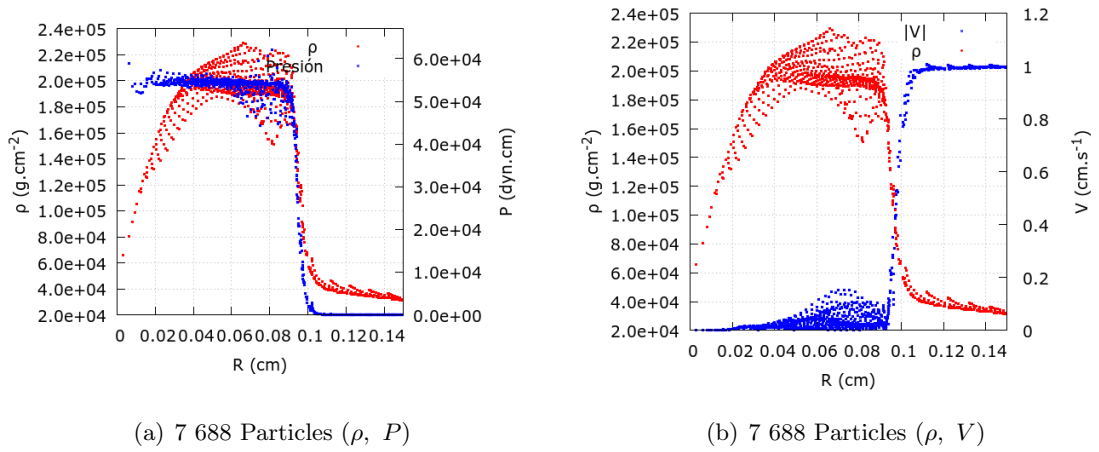


Figura 4.7: Perfiles de densidad cuando la onda de choque llega a $r < 0.1$ cm, a un tiempo de $t = 0.328$ s.

En la figura 4.8 se puede observar la evolución temporal de las partículas del modelo, y en la figura 4.9 la conservación de la energía y el momento lineal del sistema.

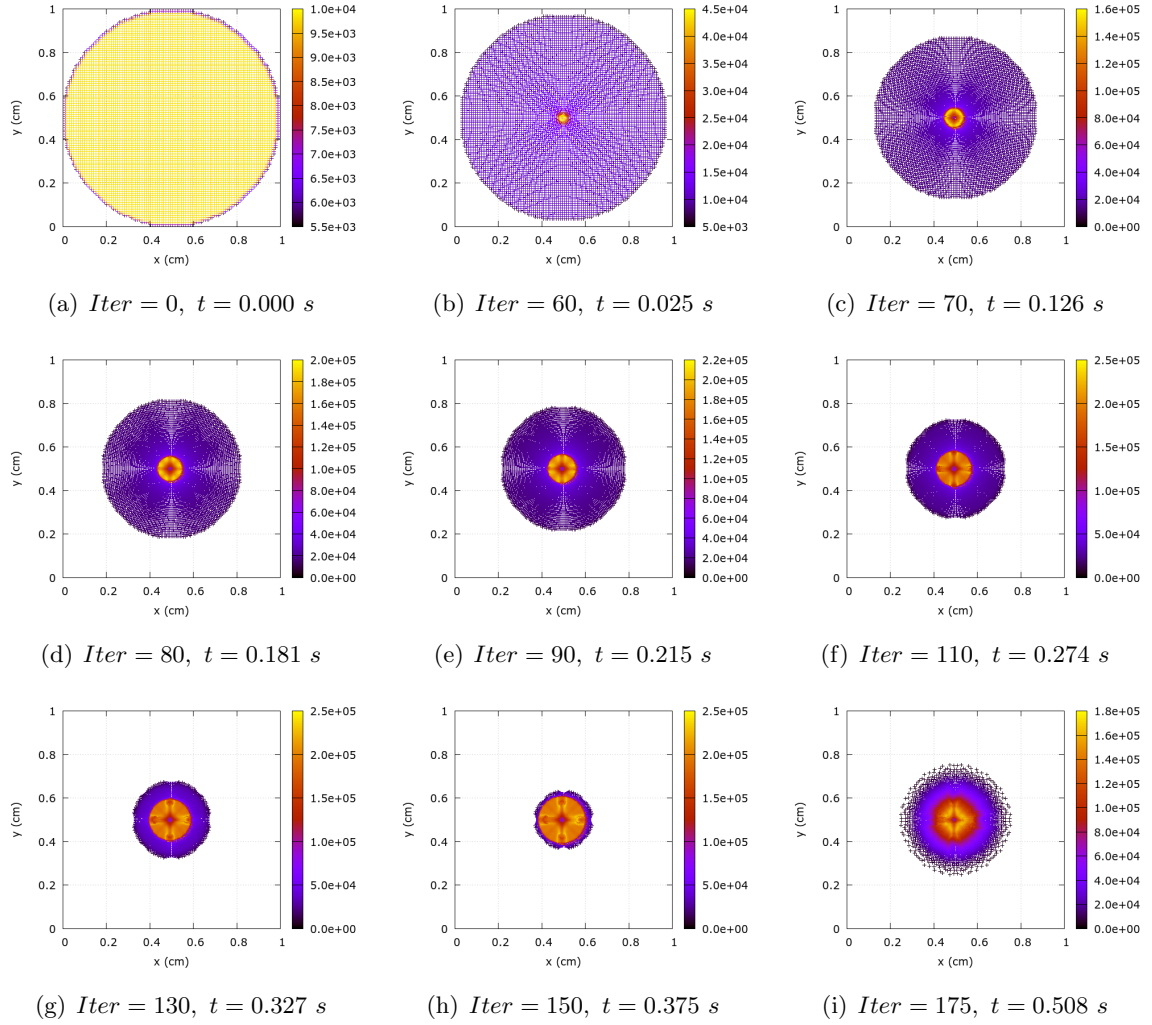


Figura 4.8: Evolución temporal de la simulación con 7 668 partículas. Se muestra el mapa de color de la densidad (g/cm^2) en diferentes instantes de tiempo.

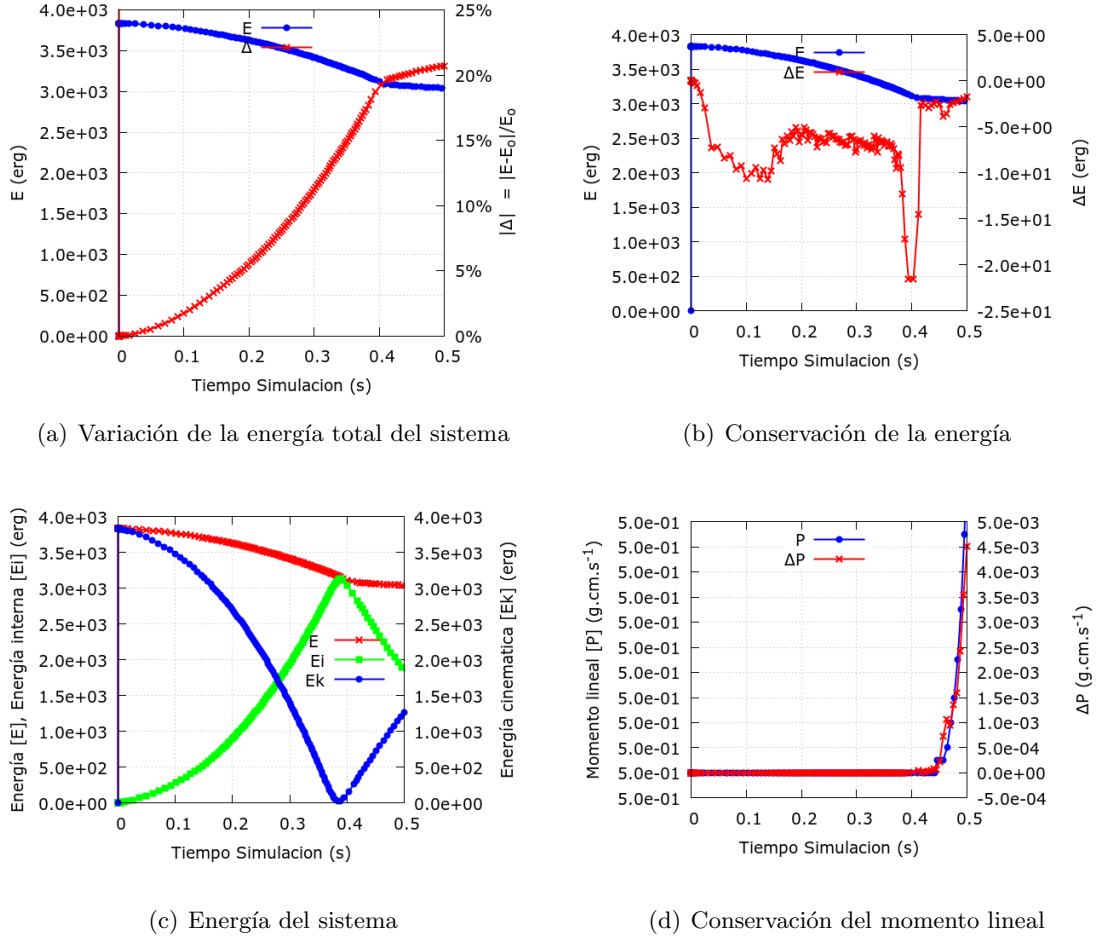


Figura 4.9: Conservación de la energía y momento lineal de la simulación con 7 668 partículas.

4.3. Inestabilidad de Rayleigh-Taylor

En este test se presenta un escenario que observa como crecen las inestabilidades hidrodinámicas durante la fase final de la compresión de una microcápsula sometida a fusión por confinamiento inercial (ICF). La última etapa de la implosión, denominada fase de estancamiento, es reconocida por su dificultad para observar las inestabilidades de Rayleigh-Taylor que son claves en la ignición (*Atzeni y Meyer-Ter-Vehn, 2004* [5], *Lindl, 1995* [50]). Existen algunos textos previos que han utilizado SPH para realizar este tipo de estudios en ICF (*Relaño y colaboradores, 1999* [74], *Relaño y colaboradores, 2006* [75]),

En el esquema habitual se simula la ignición de una cápsula de deuterio y tritio, que tiene una escala temporal de unos 10^{-10} s. Se produce una alta compresión, lanzando material al interior de la cápsula y esto genera una onda de choque que se mueve hacia el interior para conservar el momento, como una reacción a la destrucción de la parte exterior de la cápsula.

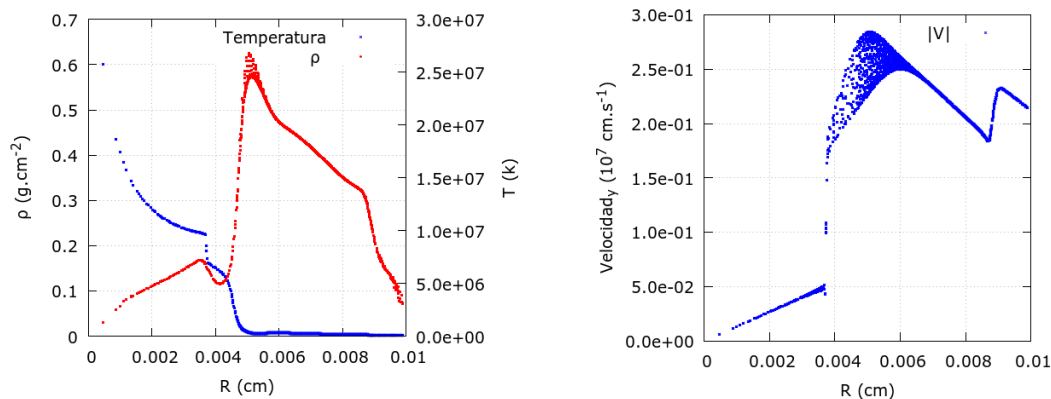


Figura 4.10: Perfil de densidad, temperatura y módulo de la velocidad (con la perturbación sinusoidal inicial incorporada) a $t = 0$ s.

Varios tipos de esquemas de ignición pueden conseguir esta implosión. En este test se utiliza el esquema estándar, donde el combustible alcanza una configuración isobárica después de la compresión adiabática inicial. En este punto tenemos dos regiones, una zona caliente central de baja densidad rodeada de una región densa y fría (figura 4.10). En un modelo más completo la fusión comenzaría en la región caliente produciendo una onda de detonación.

En la fase de estancamiento, las inestabilidades de Rayleigh-Taylor crecen cuando un fluido denso (ρ_2) es acelerado hacia otro menos denso (ρ_1), es decir cuando los gradientes de presión y densidad satisfacen $\nabla P \cdot \nabla \rho < 0$. Cuanto más negativo sea el gradiente, mas fuerte será la inestabilidad.

Si introducimos una perturbación inicial (δ_0) en la zona caliente, la perturbación crecerá de forma exponencial con una tasa de crecimiento clásica (descrita en *Chandrasekhar, 1981* [20]):

$$\delta = \delta_0 \cdot \exp(\gamma \cdot t) \quad (4.2)$$

$$\gamma = (A_t \cdot \kappa \cdot a_{int})^{1/2} \quad (4.3)$$

$$A_t = (\rho_2 - \rho_1) / (\rho_2 + \rho_1) \quad (4.4)$$

donde A_t es el número de *Atwood*, κ es el número de ondas de la perturbación y a_{int} es la aceleración en la región caliente.

Las pequeñas irregularidades iniciales rompen la simetría esférica y inducen la ampliación de las inestabilidades de Rayleigh-Taylor.

A lo largo de los años, varias simulaciones numéricas (*Atzeni, 1996* [4], *Sakaiya y colabores*

dores, 2002 [82]) han observado que la tasa de crecimiento clásica no es del todo correcta en el frente de onda, porque toma como despreciable los gradientes de densidad, la conducción térmica, etc. Para solventar este problema se han propuesto diferentes expresiones. Una de las mas usadas es la expresión modificada de *Takabe*, que algunos autores recomiendan para la simulación que aquí se muestra (*Smalyuk y colaboradores, 2002 [92]*, *Lobatchev y Betti, 2000 [51]*):

$$\gamma = \alpha \left(\frac{\kappa \cdot a_{int}}{1 + \kappa \cdot L_m} \right)^{1/2} - \beta \cdot \kappa \cdot v_{int} \quad (4.5)$$

donde L_m es el mínimo de la longitud de escala de densidad en el frente de onda ($L = \rho/\nabla\rho$), v_{int} es la velocidad en la región caliente y α, β son parámetros constantes ($\alpha \simeq 0.9$; $\beta \simeq [1 \div 4]$)

Un problema grave de esquemas numéricos al incorporar la formulación de viscosidad artificial es que introducen demasiada viscosidad en el sistema, reduciendo e incluso disipando el crecimiento de las inestabilidades (en este caso $\alpha = 1.0$ y $\beta = 2.0$). Por eso es interesante introducir las correcciones de *Balsara* (*Balsara, 1995 [7]*) para mejorar la descripción de la viscosidad artificial y que reproduzca la rarefacción esperada en los flujos de cizalla. En modelos de baja resolución, como el nuestro, la perturbación debe de ser lo suficientemente grande para que se produzcan las inestabilidades.

Modelo inicial

Las condiciones iniciales de la simulación son similares a las de *Sakagami y Nishihara, 1990 [81]* y se especifican en el cuadro 4.3.

	Radio (cm)	Densidad ($g.cm^{-2}$)	Presión (Mbar)	Velocidad ($cm.s^{-1}$)
Combustible	< 0.013	0.5	10	0
Zona de ignición	0.013 – 0.19	5.0	10	$1.5 \cdot 10^7$
Superficie	> 0.19	0.5	10	$1.5 \cdot 10^7$

Cuadro 4.3: Condiciones iniciales en 1D, que luego se trasladan a 2D, obteniendo los valores mostrados en la figura 4.10.

Se dispone de una distribución de 8 000 partículas en 2D, construida a partir de los perfiles 1D del cuadro 4.3. Para aumentar el crecimiento de las inestabilidades, se introduce un perfil

de perturbación en la velocidad, en la zona de ignición, que sigue las siguientes ecuaciones:

$$v = v_0 + \delta v \quad (4.6)$$

$$\delta v = A \cdot v_0 \cdot \sin(m \theta) \quad (4.7)$$

$$A = A_0 \cdot \exp\left(-\left(\frac{S - S_c}{\varsigma}\right)^2\right) \quad (4.8)$$

donde $S_c = 0.0469 \text{ cm}$ es la distancia a la que se produce la perturbación desde el centro de la cápsula, $m = 8$ representa el número de burbujas Rayleigh-Taylor que se van a generar y esta relacionado con el número de ondas según $\kappa = m/S_c$, $A_0 = 0.25$ y $\varsigma = 5 \cdot 10^{-4} \text{ cm}$ son parámetros constantes que determinan la amplitud de la perturbación y el ancho de función *Gaussiana* alrededor de S_c donde se aplica la perturbación.

Resultados

El cuadro 4.4 describe el comportamiento de la ejecución del código.

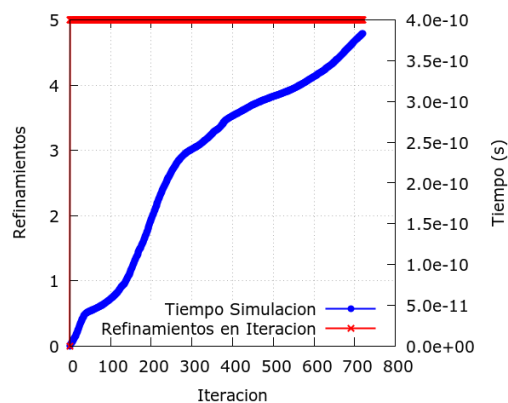
Parts.	PC	$\tau_{Model} \text{ (ns)}$	$\tau_{CPU_{system}} \text{ (h)}$	$\tau_{CPU_{user}} \text{ (h)}$	Speedup	Iters.	Refs.
8 000	A	0.274	32.75	9.75	3.36	378	3 070

Cuadro 4.4: Tiempo de computación del código implícito. Los valores son obtenidos cuando la onda de choque llega al borde del sistema simulado en $r = 0.01 \text{ cm}$ desde el centro de masas. La ejecución en el PC A se realiza con 8 threads en un portatil con CPU Intel Core i7-2720QM 2.2Ghz y 16 GB DDR3-1600Mhz.

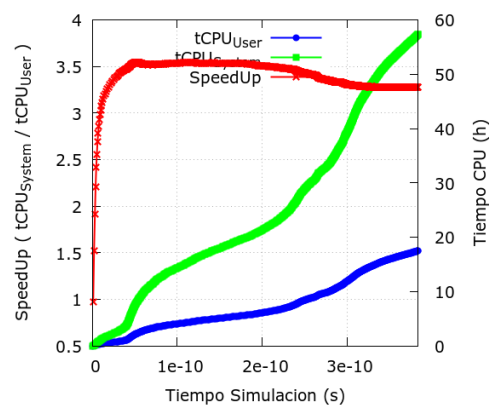
En la figura 4.11 se muestran los tiempos de computación.

En la figura 4.12 se muestran los perfiles de densidad y velocidad de forma simultánea, cuando la onda de choque llega al borde exterior de la simulación.

En la figura 4.13 se puede observar la evolución temporal de las partículas del modelo, y en la figura 4.14 la conservación de la energía y el momento lineal del sistema. En la figura 4.13(b), se ve claramente el caracter sinusoidal de la perturbación de la velocidad. En los gráficos siguientes se aprecia la asimetría entre los picos más densos y las burbujas mucho menos densas. Si la



(a) Iteración Vs Refinamientos



(b) Speedup

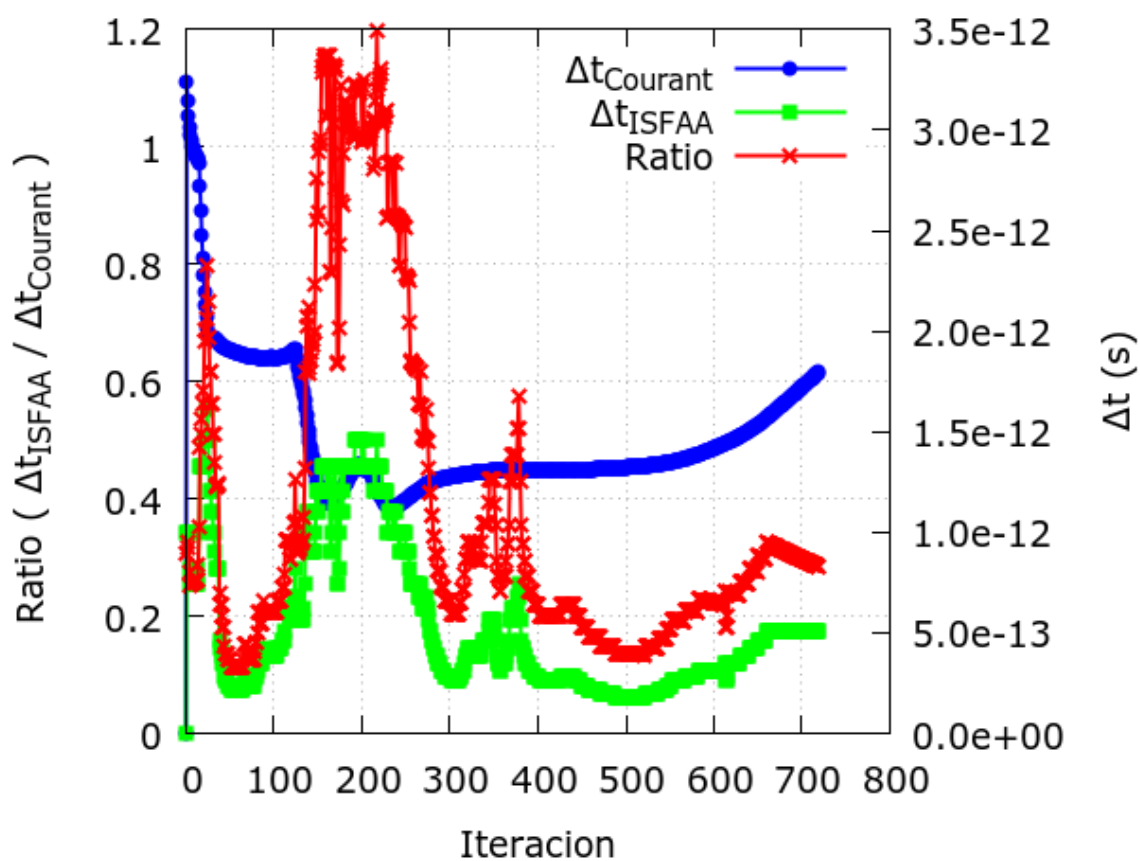
(c) Ratio $\Delta t_{ISFAA} / \Delta t_{Courant}$

Figura 4.11: Tiempos de computación de la simulación con 7 668 partículas.

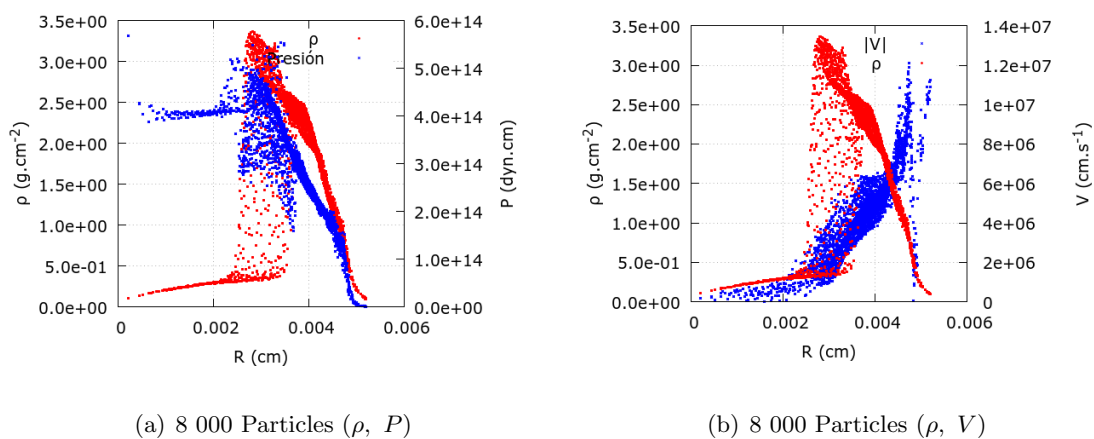


Figura 4.12: Perfiles de densidad cuando la onda de choque llega al borde exterior de la simulación, a un tiempo de $t = 0.274$ ns.

resolución fuera mayor, podríamos observar que en los lados de las burbujas, se producirían los vórtices característicos de las inestabilidades de Kelvin-Helmholtz.

De las pruebas realizadas se determina que el sistema es muy dependiente de la geometría de la perturbación inicial y, para el crecimiento de las inestabilidades, de la resolución que del sistema. Si aumentamos m tenemos más perturbaciones, lo que en la práctica representa una reducción local de la resolución.

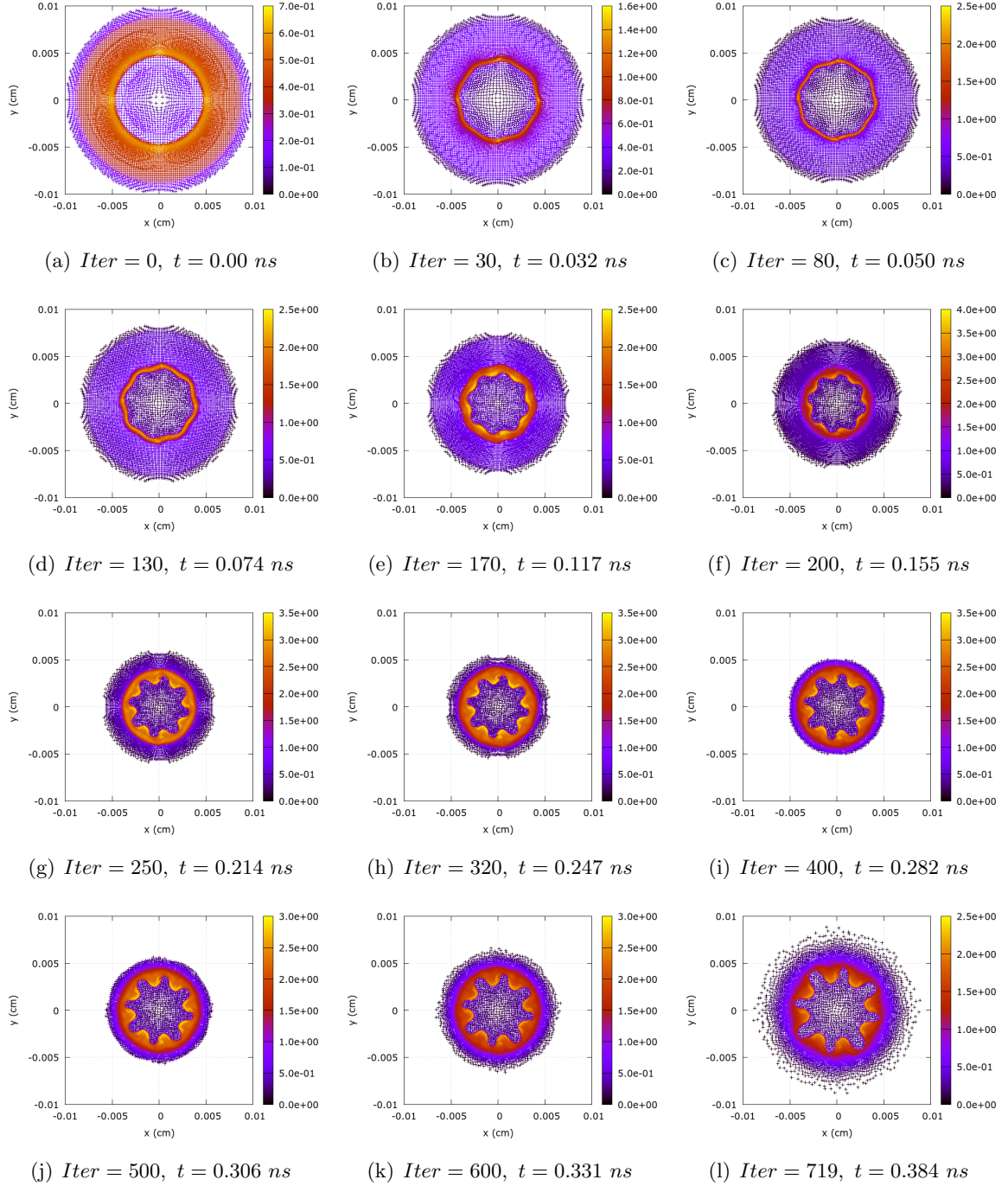


Figura 4.13: Evolución temporal de la inestabilidad de Rayleigh-Taylor en la zona de estancamiento del ICF con 8 000 partículas, con $m = 8$. Se muestra el mapa de color de la densidad (g/cm^2) en diferentes instantes de tiempo.

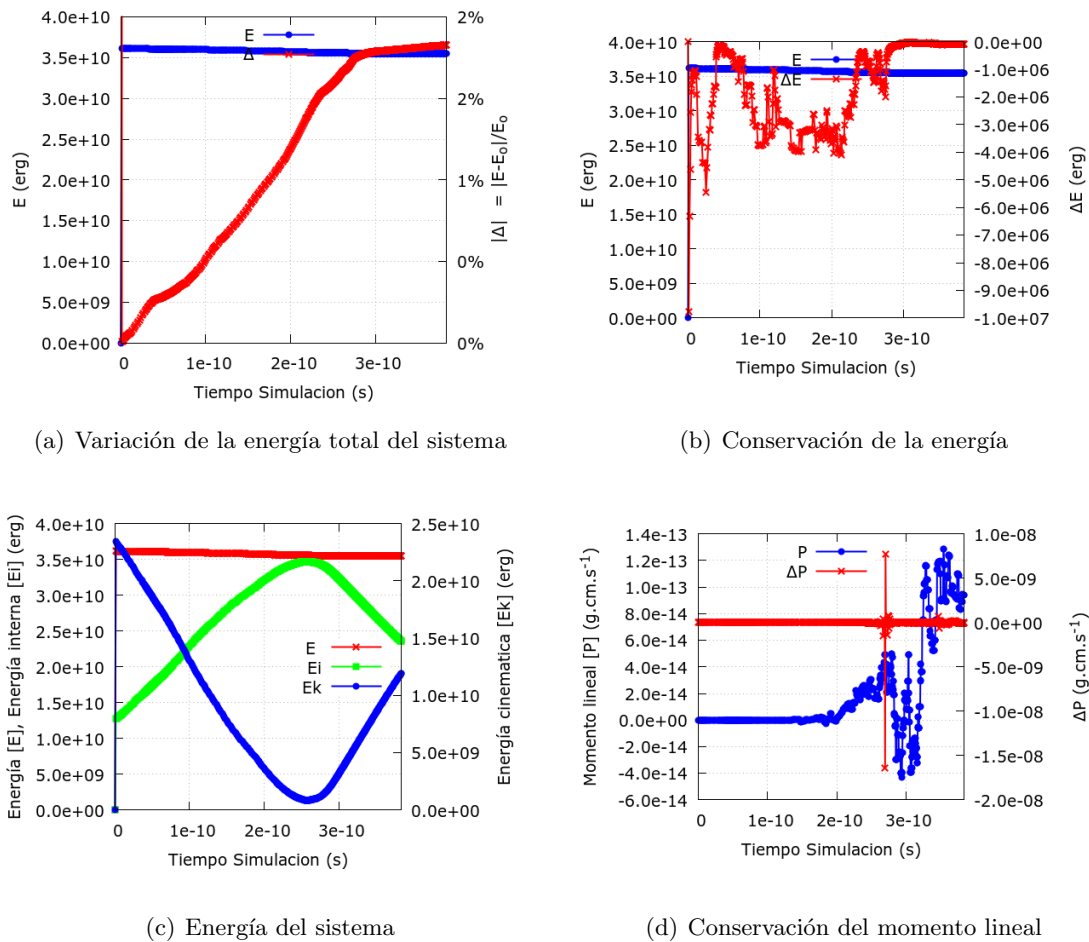


Figura 4.14: Conservación de la energía y momento lineal de la simulación con 8 000 partículas.

4.4. Caída libre

El test de la caída libre consiste en el colapso gravitatorio de una esfera gaseosa.

Para simular el colapso, la presión del gas y la viscosidad artificial se anulan de modo que la estructura se derrumba bajo la fuerza de la gravedad. La implosión del sistema se deja evolucionar, a energía constante, hasta que el tiempo transcurrido se encuentra cerca del tiempo máximo de colapso t_{col} .

Si tenemos una esfera de densidad uniforme con masa M y radio R , la partícula más alejada del centro de masas tendrá un tiempo característico de colapso, el máximo para todo el sistema, descrito por:

$$t_{col} = \frac{\pi}{2} \left(\frac{R^3}{2 G M} \right)^{1/2} \quad (4.9)$$

Para una partícula cualquiera i la solución analítica de la caída libre sigue la ecuación:

$$\frac{t}{t_{col}} = 1 - \frac{2}{\pi} \left(\sin^{-1} \left(\frac{r}{r_0} \right)^{1/2} - \left(\frac{r}{r_0} \right)^{1/2} \left(1 - \frac{r}{r_0} \right)^{1/2} \right) \quad (4.10)$$

donde r_0 es su radio inicial y r es el radio de i a tiempo t .

La simetría esférica del test de la caída libre es muy exigente porque la evolución es altamente no lineal, lo que permite una buena comprobación de la gravedad. Es por ello que se aprovecha este test para comprobar las diferentes implementaciones de la gravedad (sección 2.2.4):

1. Gravedad exacta: Calculada de forma directa explícita, que tiene un coste de $\Theta(n^2)$.
2. Gravedad calculada con la ley de Gauss, que solamente sirve para modelos con simetría esférica y que tiene un coste de $\Theta(n \log n)$.
3. Gravedad calculada con la aproximación multipolar del potencial gravitatorio, que tiene un coste de $\Theta(n \log n)$.

Modelo inicial

Los modelos que aquí se muestran son los siguientes:

- Modelo 2D con 3 720 partículas, distribuidas en una forma circular con la gravedad según la ley de Gauss para un cilindro 2D.
- Modelo 3D con 9 134 partículas, distribuidas en una forma esférica con la gravedad según la ley de Gauss para una esfera 3D.

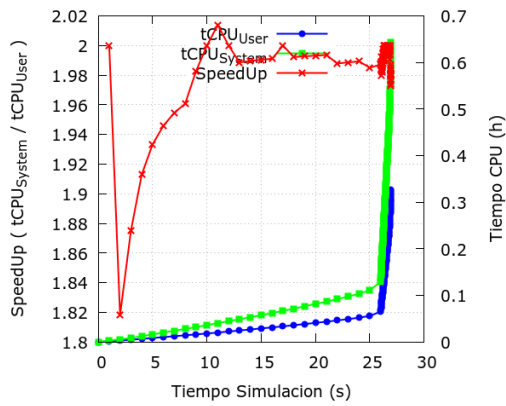
Se anula la presión de las partículas en la ecuación de estado, $P = 0 \text{ dyn.cm}^{-3}$ y no se tiene en cuenta la viscosidad artificial.

Para poder mostrar mejor la evolución, sobre todo de las fases finales del colapso, se ha tomado un paso de tiempo fijo de $\Delta t = 10e^{-2}$.

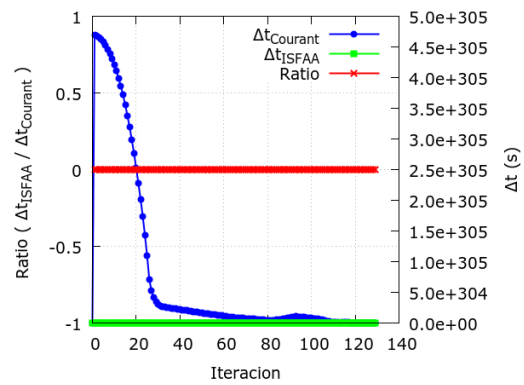
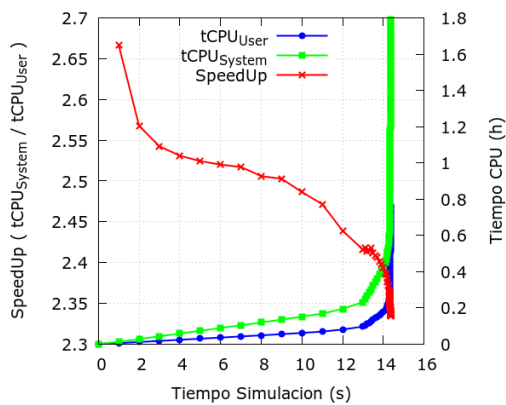
Resultados

El cuadro 4.5 describe el comportamiento de la ejecución del código para los dos modelos.

En la figura 4.15 se muestran los tiempos de computación de los dos modelos.



(a) Modelo 2D - Speedup

(b) Modelo 2D - Ratio $\Delta t_{ISFAA} / \Delta t_{Courant}$ 

(c) Modelo 3D - Speedup

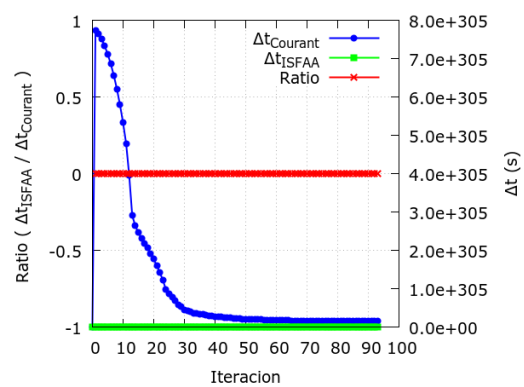
(d) Modelo 3D - Ratio $\Delta t_{ISFAA} / \Delta t_{Courant}$

Figura 4.15: Tiempos de computación de la simulación, modelos 2D con 3 720 partículas y en 3D con 9 134. Se observa que, a pesar de ser modelos con diferentes dimensiones, a mayor número de partículas mayor es el speedup debido que el cálculo paralelo de la inversión de la matriz requiere mayor tiempo.

Tipo de cálculo de la gravedad	Parts.	PC	$\tau_{Model} (s)$	$\tau_{CPU_{system}} (h)$	$\tau_{CPU_{user}} (h)$	Speedup	Iters.
Aprox. con ley de gauss 2D	3 720	A	26.91	0.64	0.33	1.939	129
Aprox. con ley de gauss 3D	9 134	A	14.34	1.79	0.77	2.325	93

Cuadro 4.5: Tiempo de computación del código implícito en función de los diferentes cálculos de la gravedad. La ejecución en el PC A se realiza con 8 threads en un portatil con CPU Intel Core i7-2720QM 2.2Ghz y 16 GB DDR3-1600Mhz.

En la figura 4.16 se muestran los perfiles de densidad, velocidad y gravedad, cuando la implosión esta próxima al centro de masas inicial para los dos modelos.

En las figuras se puede observar la evolución temporal del modelo 2D con 3 720 partículas (figura 4.17) y en el modelo 3D con 9 134 partículas (figura 4.18).

En las figuras se puede observar la conservación de la energía y el momento lineal del sistema del modelo 2D con 3 720 partículas (figura 4.19) y en el modelo 3D con 9 134 partículas (figura 4.20).

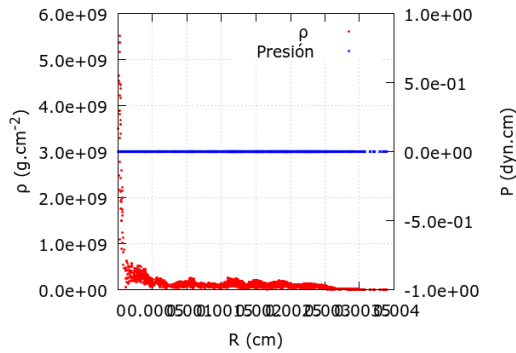
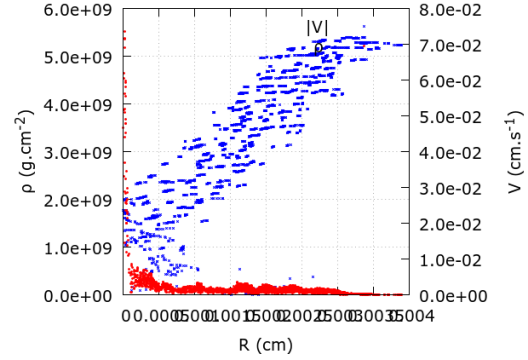
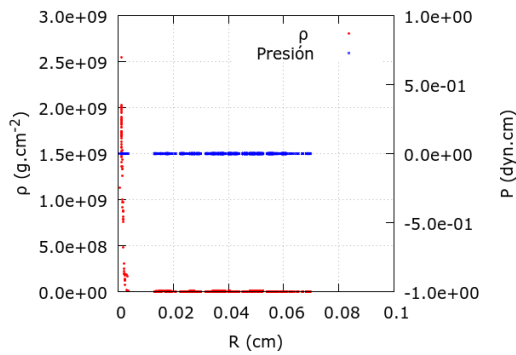
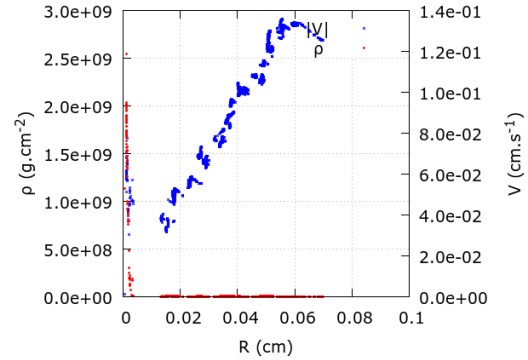
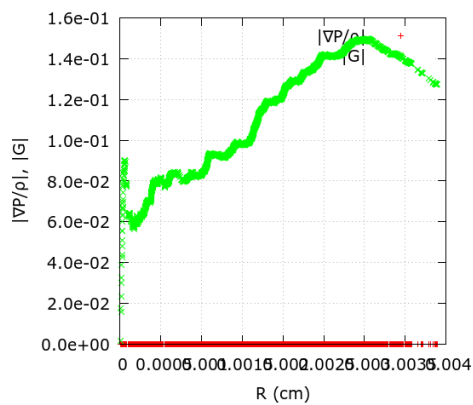
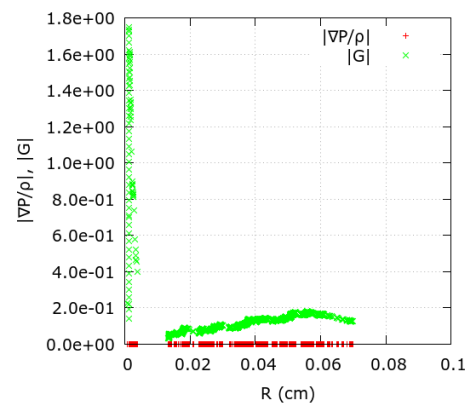
(a) 2D con 3 720 (ρ , P), Iter=129(b) 2D con 3 720 (ρ , V), Iter=129(c) 3D con 9 134 (ρ , P), Iter=93(d) 3D con 9 134 (ρ , V), Iter=93(e) 2D con 3 720 ($|\nabla P/\rho|$ Vs $|G|$), Iter=129(f) 3D con 9 134 ($|\nabla P/\rho|$ Vs $|G|$), Iter=93

Figura 4.16: Perfiles de densidad cuando la implosión esta próxima al centro de masas inicial. En los paneles 4.16(e) y 4.16(f) podemos observar la gravedad del sistema.

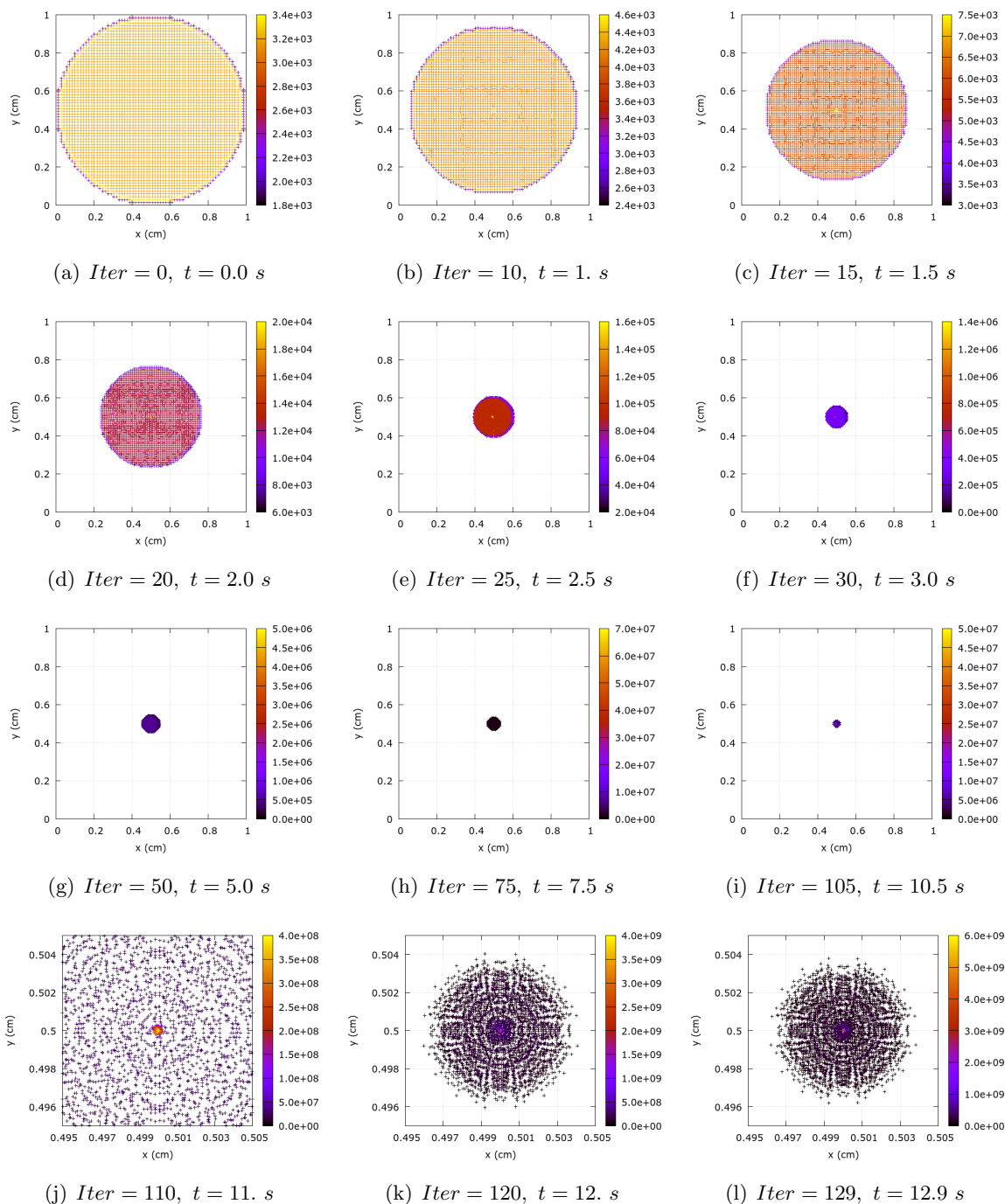


Figura 4.17: Evolución temporal del modelo 2D con 3 720 partículas. los paneles 4.17(j) 4.17(k) y 4.17(l) tienen una escala mas cercana al centro de masas.

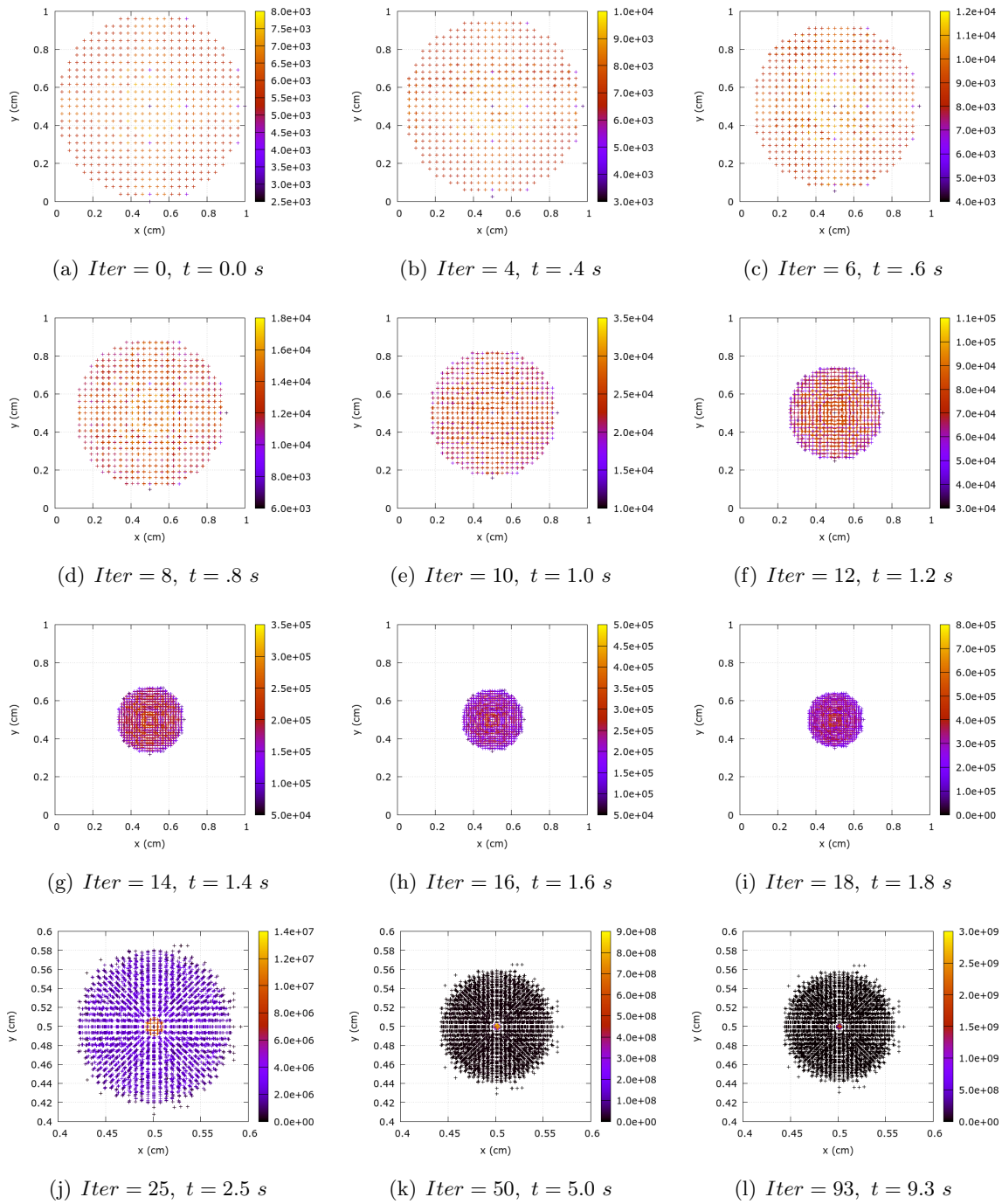
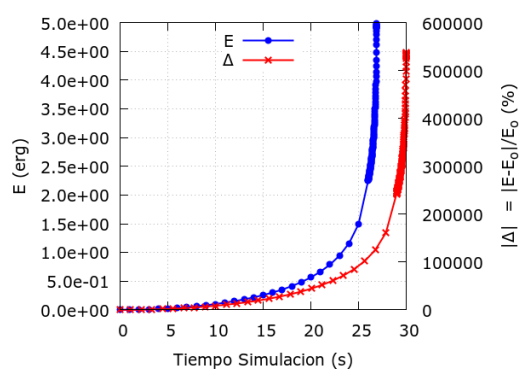
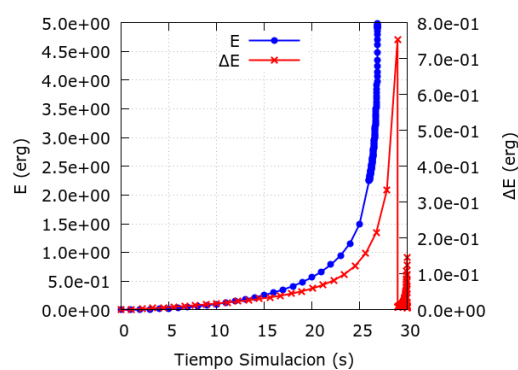


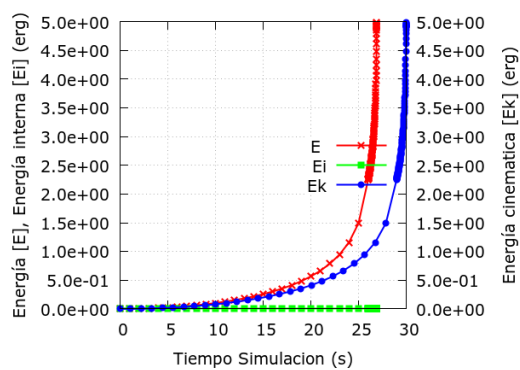
Figura 4.18: Evolución temporal del modelo 3D con 9134 partículas. los paneles 4.18(j) 4.18(k) y 4.18(l) tienen una escala mas cercana al centro de masas.



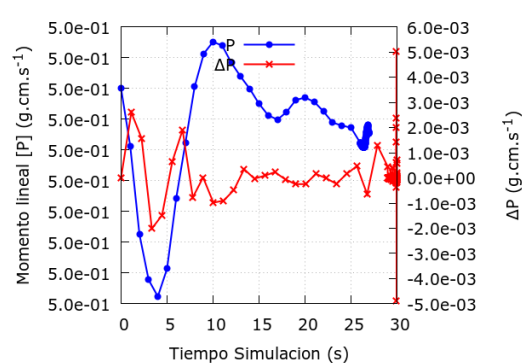
(a) Variación de la energía total del sistema



(b) Conservación de la energía

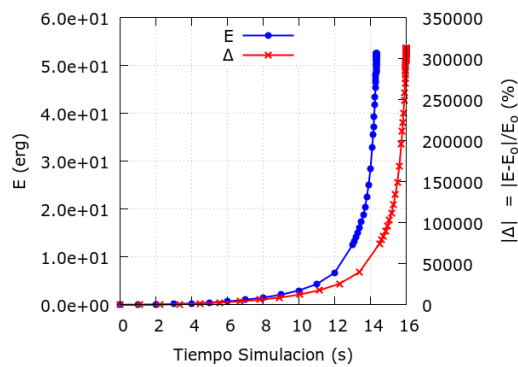


(c) Energía del sistema

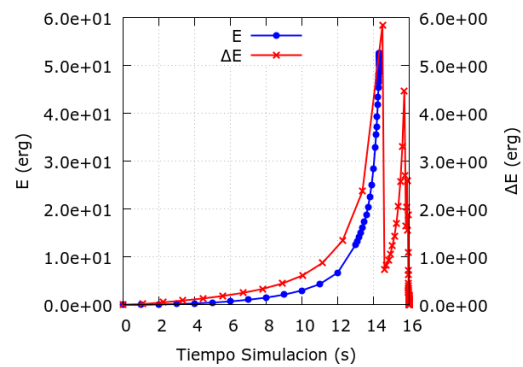


(d) Conservación del momento lineal

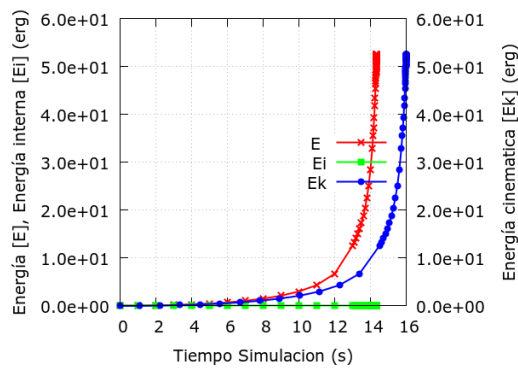
Figura 4.19: Conservación de la energía y momento lineal de la simulación con el modelo 2D de 3 720 partículas.



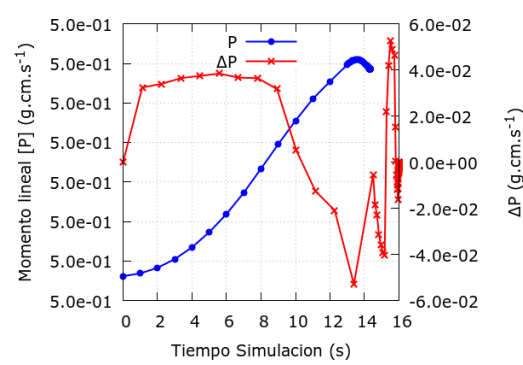
(a) Variación de la energía total del sistema



(b) Conservación de la energía



(c) Energía del sistema



(d) Conservación del momento lineal

Figura 4.20: Conservación de la energía y momento lineal de la simulación con el modelo 3D de 9 134 partículas.

4.5. Evolución de una onda térmica

Este test consiste en observar la evolución de una onda térmica a través de un medio homogéneo a una densidad constante, según se describe en *Cabezón y colaboradores, 2008* [17]. En la evolución del modelo se asume que las partículas no se mueven y, por tanto, la evolución de la onda esta determinada únicamente por la ecuación de la energía. El objetivo es determinar en que medida el código es capaz de reproducir la difusión de calor ajustándose a la solución analítica descrita por *Zel'dovich y Raizer, 1967* [104].

La evolución de la onda térmica que se desplaza en el interior de un medio estático se encuentra determinada por la ecuación de la energía, Eq. (2.133):

$$\frac{\partial U_i}{\partial t} = \frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \left(\frac{dU_i}{dt} \right)_{visco} + \left(\frac{dU_i}{dt} \right)_{cond}$$

esta ecuación puede ser simplificada al tener solo conductividad térmica, Eq. (2.125):

$$\left(\frac{dU_i}{dt} \right)_{cond} = \sum_{j=1}^N \frac{m_j}{\rho_i \rho_j} \frac{(\kappa_i + \kappa_j) (T_j - T_i)}{r_{ij}^2} \mathbf{r}_{ij} \cdot \nabla_i \mathbf{W}_{ij}$$

donde κ es el coeficiente de conductividad térmica.

Otro punto interesante se encuentra en las duras condiciones iniciales que tratan de representar una discontinuidad térmica. Las dificultades que se observan para manejar estas discontinuidades térmicas son similares al caso de los choques hidrodinámicos multidimensionales, y por tanto, es necesario ampliar de forma artificial la resolución del código.

Una aplicación de este test a un caso real lo encontramos en *Hillebrandt y Niemeyer, 2000* [39], donde una onda térmica, con un grosor pequeño de 1 cm, es el precursor de una ignición nuclear que termina con la explosión de una supernova de tipo Ia.

Para el modelo inicial se ha escogido la implementación descrita por *Jubelgas y colaboradores, 2004* [44]. que establecen la distribución de energía inicial siguiendo la función de *Green*, que es solución exacta al problema de la conducción para la siguiente función inicial:

$$U(r, t) = \frac{A}{4\pi \alpha t} \exp\left(-\frac{r^2}{4\alpha t}\right) + U_o \quad (4.11)$$

donde el parámetro α representa la difusividad térmica a volumen constante, tiene unidades de $\text{cm}^3 \cdot \text{s}^{-1}$, y según se cita en *Zel'dovich y Raizer, 1967* [104], responde a la siguiente ecuación:

$$\alpha = \frac{\kappa}{\rho C_v} \quad (4.12)$$

donde el parámetro κ es la conductividad térmica, con un valor de $\kappa_i = \bar{\rho}_{ij} \cdot \bar{c}_{vij} \cdot (\bar{h}_{ij} \cdot \bar{c}_{ij} + 4|\mu_{ij}|)$ y que se encuentra definida en la Eq. (2.118); C_v es el calor específico a volumen constante.

El ratio de cambio de la energía térmica se puede obtener derivando la ecuación 4.11 con respecto al tiempo:

$$\frac{d U(r, t)}{dt} = \frac{A}{4\pi\alpha t^2} \left[\exp\left(-\frac{r^2}{4v\alpha t}\right) \right] \left(\frac{r^2}{4\alpha t} - 1 \right) \quad (4.13)$$

Modelo inicial

Se dispone de un modelo circular en 3D con 4 148 partículas distribuidas de forma equidistante. Dado un tiempo inicial $t = 0.4 \text{ s}$, que después se toma como tiempo inicial, la ecuación 4.11 proporciona el perfil preciso de una onda térmica que emerge de la discontinuidad inicial. En este test se toman como condiciones iniciales: $A = 10^5 \text{ erg.cm}^2.\text{g}^{-1}$, $U_o = 10^3 \text{ erg.g}^{-1}$, $r_{max} = 8 \text{ cm}$ y $\rho = 1 \text{ g.cm}^{-2}$.

Solo la ecuación de la temperatura se coloca en el esquema y por tanto, solo se modifica la temperatura de las partículas y no su posición. El paso de tiempo se ha tomado fijo a $\Delta t = 1.0 \text{ s}$.

Resultados

En la figura 4.21 se puede observar la posición de las partículas del modelo, proyectado en un plano 2D en la última iteración de la simulación, la posición no varía durante la evolución.

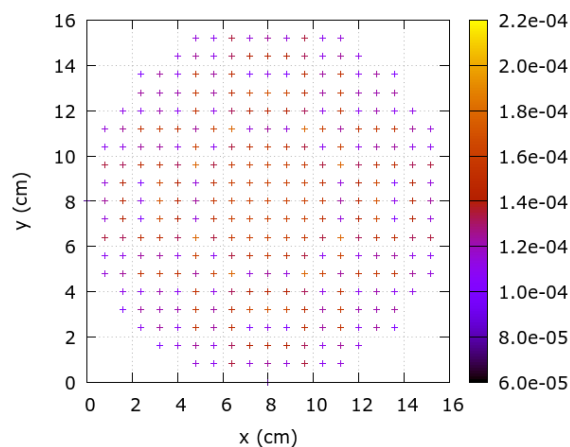


Figura 4.21: Posiciones de las partículas en la simulación con 4 148 partículas a $t = 50.4 \text{ s}$, Iter=50. Se muestra el mapa de color de la densidad (g/cm^3).

En la figura 4.22 podemos observar la evolución temporal de la temperatura hasta que la onda térmica esta prácticamente disipada.

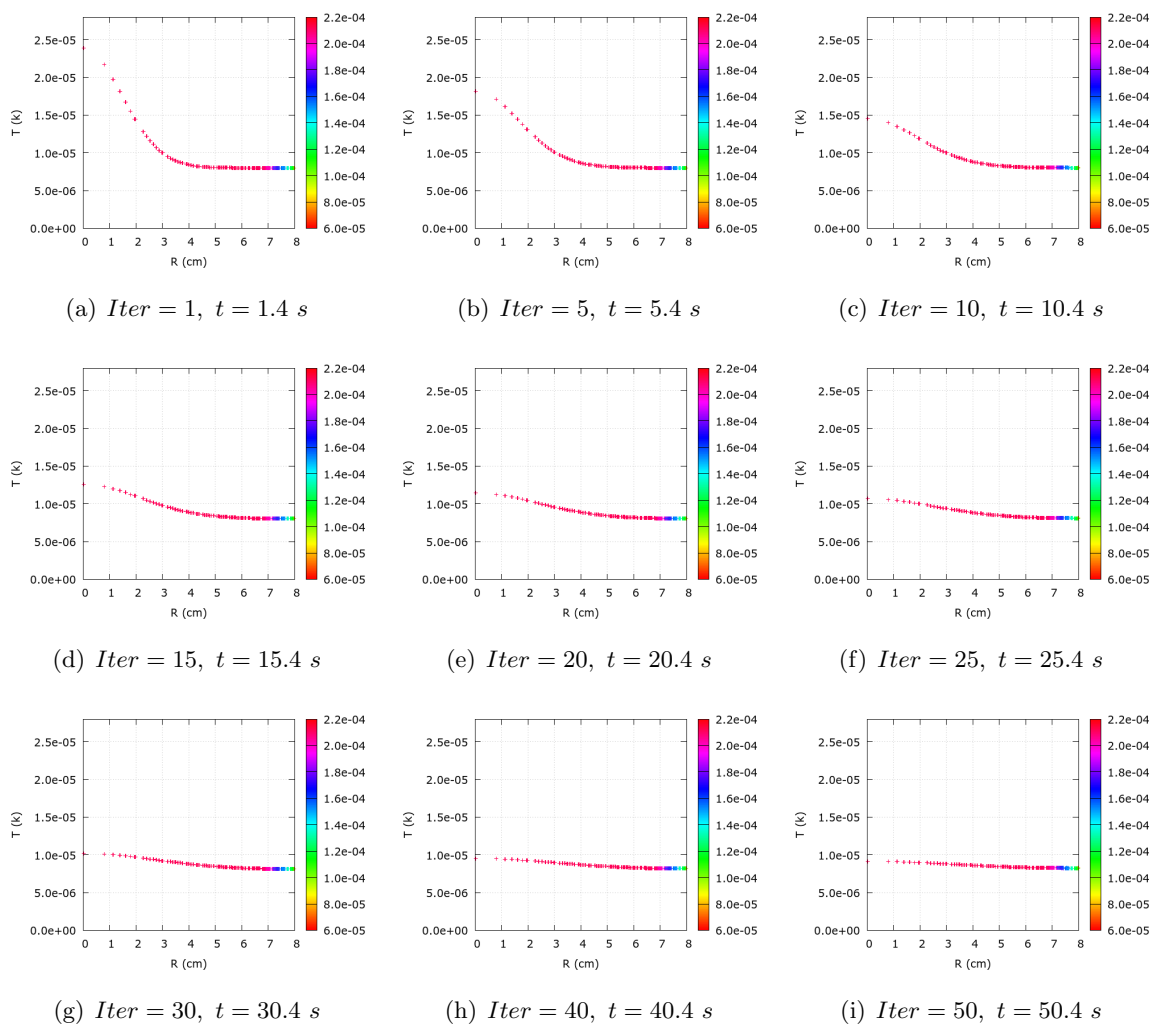


Figura 4.22: Evolución temporal de la temperatura en la simulación con 4148 partículas. Se muestra el mapa de color de la densidad (g/cm^3) en diferentes instantes de tiempo.

Capítulo 5

Tests con gravedad

Una vez que se ha validado el código ISFAA con los test, se aplica a escenarios propiamente astrofísicos. Para ello será necesario activar toda la física implementada.

Como en el caso de los tests, se establece por cada aplicación el siguiente esquema:

- Descripción: Donde se analiza lo que se va a probar y lo que se espera obtener.
- Modelo inicial: Donde se da una descripción física de las variables del modelo.
- Resultados: Donde se analizan las salidas de la simulación.

Para obtener resultados en los tests con gravedad que simulen los casos reales es necesario crear modelos en 3D. Debido al coste computacional del método implícito esto es todo un hándicap. Para obtener resultados precisos en 3D, que describan perfectamente las ondas de rarefacción, es necesario al menos unas 100 000 partículas, pero con la capacidad de cálculo de la que se dispono (servidor multinúcleo de 8 CPUs con tecnología *HyperThreading* y 48GB de RAM) solamente modelos de un máximo de 30 000 partículas en 3D pueden ser calculados. A pesar de ello los resultados siguen siendo válidos, simplemente la resolución es menor y, por tanto, la interpolación o suavizado entre puntos es mayor, disminuyendo la precisión.

Los tests con gravedad que se han simulado son las siguientes:

1. Evolución casi-estática de un sistema próximo al equilibrio (Toy Star en 2D).
2. Equilibrio hidrostático de una estrella de tipo solar, construido con un polítropo.
3. Equilibrio hidrostático de una enana blanca masiva de $1.15M_{\odot}$.

4. Aproximación implícita cuasi-hidrostática. Esto supone la evolución de un sistema casi-estático, en el que las velocidades no se encuentran explícitamente en el modelo y está orientado a demostrar que puede evolucionar en largos intervalos temporales.

5.1. Evolución casi-estática de un sistema próximo al equilibrio

Basado en las estrellas de juguete descritas por *Monaghan, 2005* [61], se genera un sistema que se sostiene por una gravedad más simple que la real, específica para este tipo de modelos.

Se observará que el sistema consigue un equilibrio con simetría esférica después de varias pulsaciones.

Gravedad basada en un oscilador armónico

La gravedad de estos sistemas próximos al equilibrio es una simplificación de un oscilador armónico, descrito en el apéndice C.2, donde la gravedad tiene un valor proporcional a la distancia al centro de masas.

Para ello, tomamos un valor de $\kappa = 1$ y $n = 1$, obteniendo las ecuaciones siguientes:

$\mathbf{g}^i = -\mathbf{r}_i^{CM} = -r_i^{CM} \cdot \widehat{\mathbf{r}_i^{CM}}$
$g^{x_i} = -r_i^{CM} \cdot \frac{x_i - x^{CM}}{r_i^{CM}} = -(x_i - x^{CM})$
$g^{y_i} = -r_i^{CM} \cdot \frac{y_i - y^{CM}}{r_i^{CM}} = -(y_i - y^{CM})$
$g^{z_i} = -r_i^{CM} \cdot \frac{z_i - z^{CM}}{r_i^{CM}} = -(z_i - z^{CM})$

$\frac{\partial g^{x_i}}{\partial x_i} =$	-1	$\frac{\partial g^{y_i}}{\partial x_i} =$	0	$\frac{\partial g^{z_i}}{\partial x_i} =$	0
$\frac{\partial g^{x_i}}{\partial y_i} =$	0	$\frac{\partial g^{y_i}}{\partial y_i} =$	-1	$\frac{\partial g^{z_i}}{\partial y_i} =$	0
$\frac{\partial g^{x_i}}{\partial z_i} =$	0	$\frac{\partial g^{y_i}}{\partial z_i} =$	0	$\frac{\partial g^{z_i}}{\partial z_i} =$	-1

Modelo inicial

Se dispone de un modelo en 2D con 8 000 partículas distribuidas a partir de un grid equidistante al que se realiza “stretching” para obtener el perfil de densidad adecuado. Este perfil procede de una integración analítica descrita por la ecuación:

$$\rho = \rho_c \cdot e^{-(r^2/2K)} \quad (5.1)$$

donde el valor de $K = (\gamma - 1) U_o$.

Si en la simulación tomamos la energía inicial como $U_o = 3$, obtenemos un valor de $K = 2$ y el perfil de la densidad queda como sigue:

$$\rho = \rho_c \cdot e^{-(r^2/4)} \quad (5.2)$$

donde el valor de la densidad central que se ha tomado es $\rho_c = 1 \text{ g.cm}^{-2}$.

Resultados

El cuadro 5.1 describe el comportamiento de la ejecución del código.

En la figura 5.1 se muestran los tiempos de computación.

En la figura 5.2 se muestran los perfiles cuando el modelo llega a la estabilidad.

En la figura 5.3 se puede observar la evolución temporal de las partículas del modelo.

En la figura 5.4 se puede observar la conservación de la energía del sistema.

En la figura 5.5 se puede observar la conservación del momento lineal y angular del sistema.

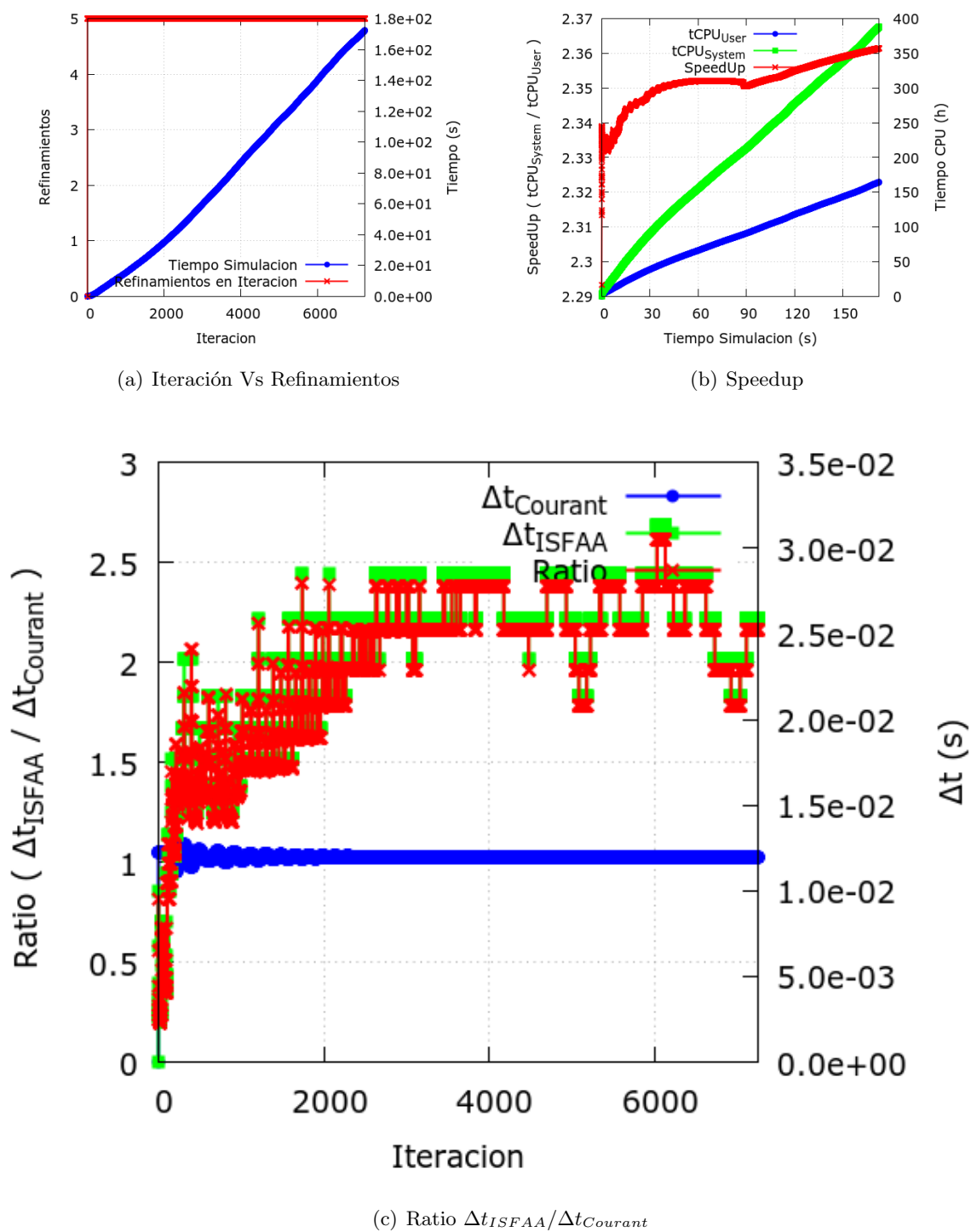


Figura 5.1: Tiempos de computación de la simulación con 8 000 partículas.

Parts.	PC	$\tau_{Model} (s)$	$\tau_{CPU_{system}} (d)$	$\tau_{CPU_{user}} (d)$	Speedup	Iters.
8 000	C	172.5	16.18	6.85	2.36	7240

Cuadro 5.1: Tiempo de computación del código implícito. Los valores son obtenidos cuando el modelo se estabiliza. La ejecución en el PC C se realiza con 4 threads en un PC de sobremesa con CPU Intel Core 2 Quad-8400Q 2.33Ghz y 6 GB DDR2-667Mhz. El tiempo de computación, tanto de sistema como de usuario, se especifica en días.

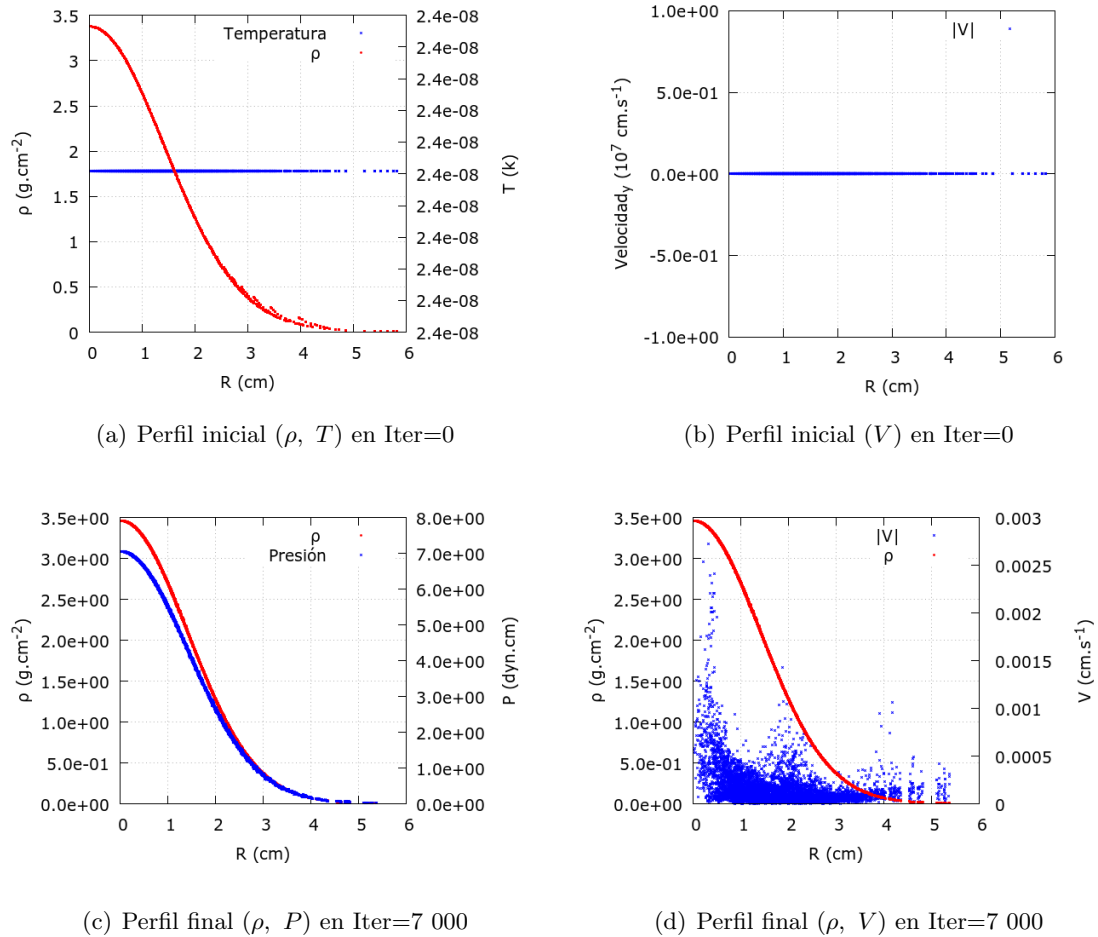


Figura 5.2: Perfiles de densidad, velocidad, presión y temperatura cuando la simulación consigue la estabilidad. En el panel 5.2(d) se observa que el modelo se encuentra prácticamente sin velocidad.

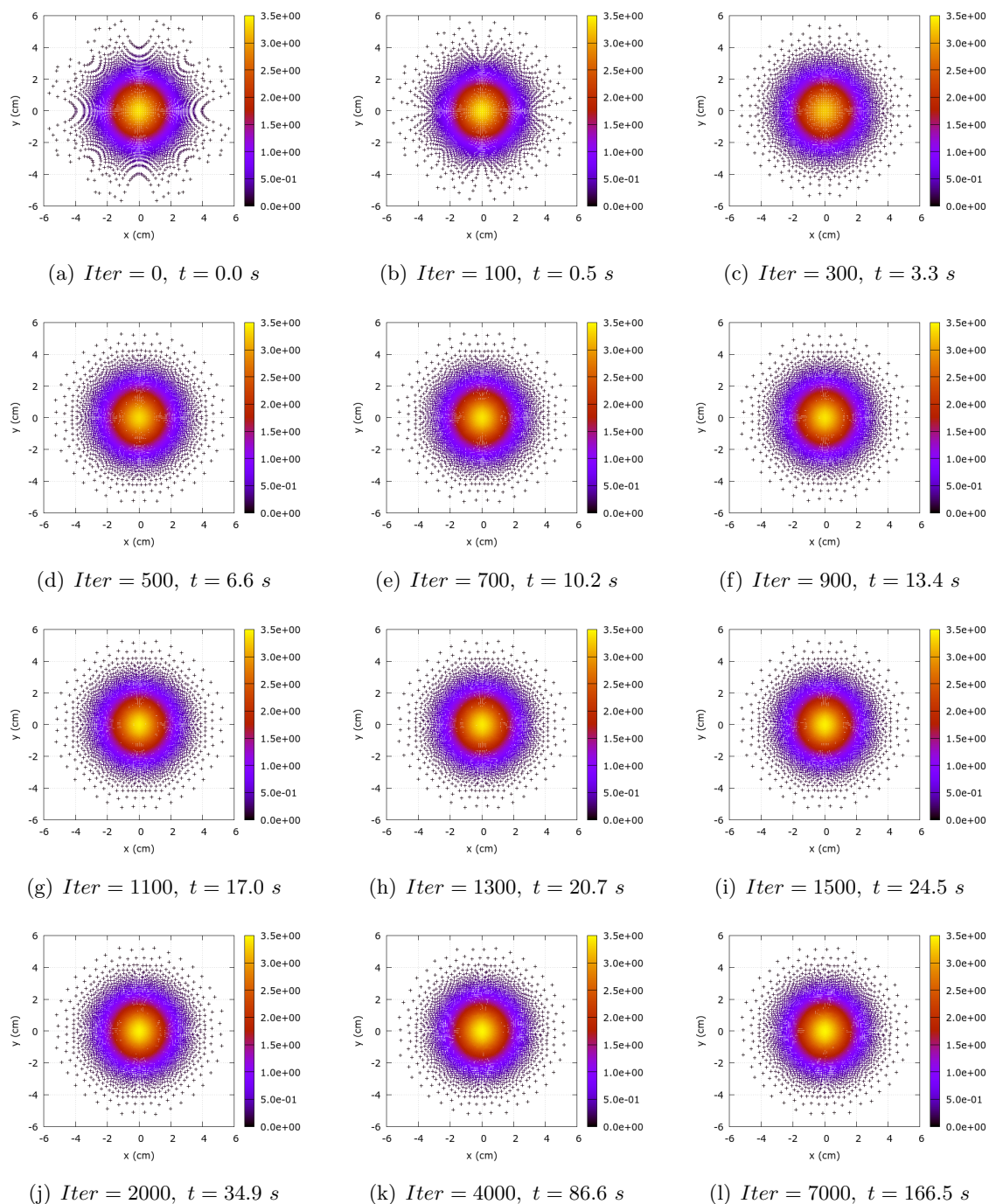


Figura 5.3: Evolución temporal del modelo de evolución cuasi-estática de un sistema próximo al equilibrio. Se muestra el mapa de color de la densidad (g/cm^2) en diferentes instantes de tiempo.

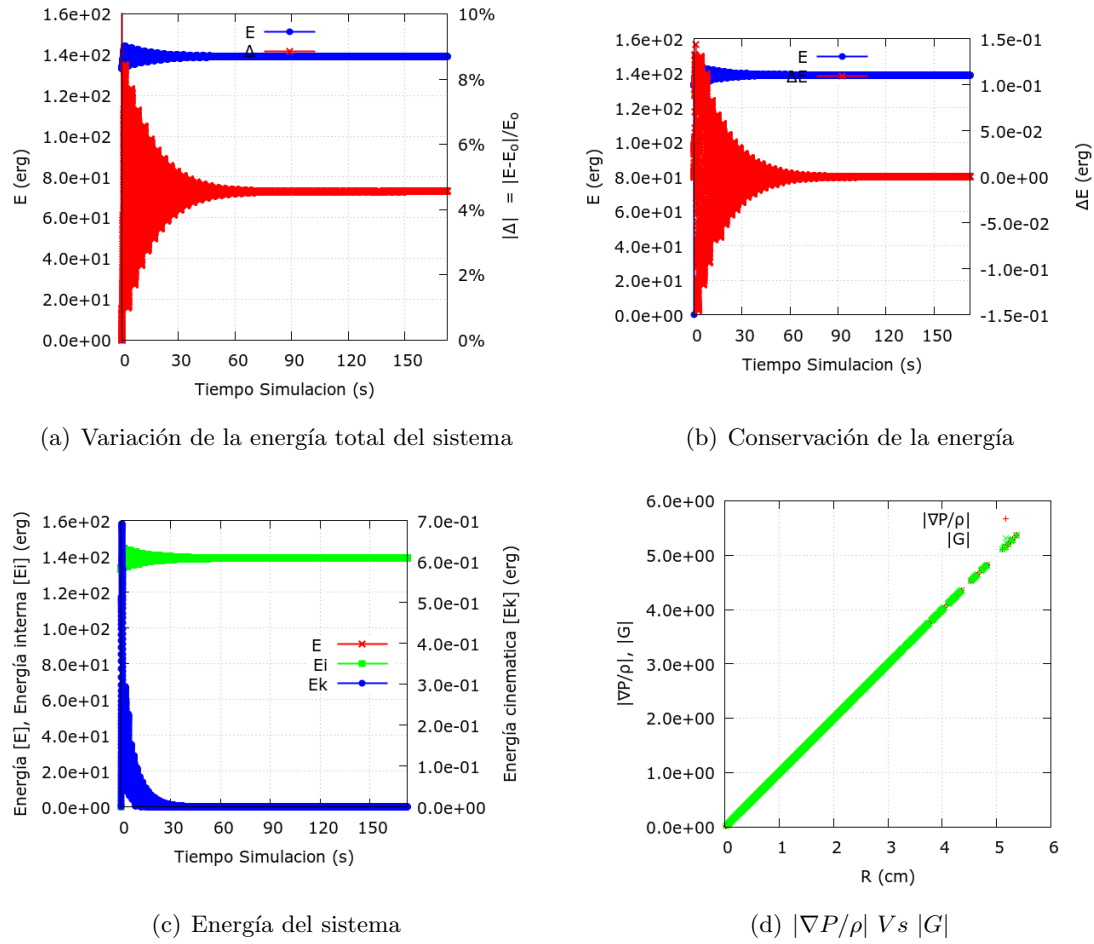


Figura 5.4: Conservación de la energía en la simulación y equilibrio del sistema. En el panel 5.4(d) se puede observar el ajuste con la gravedad proporcional a la distancia en el equilibrio.

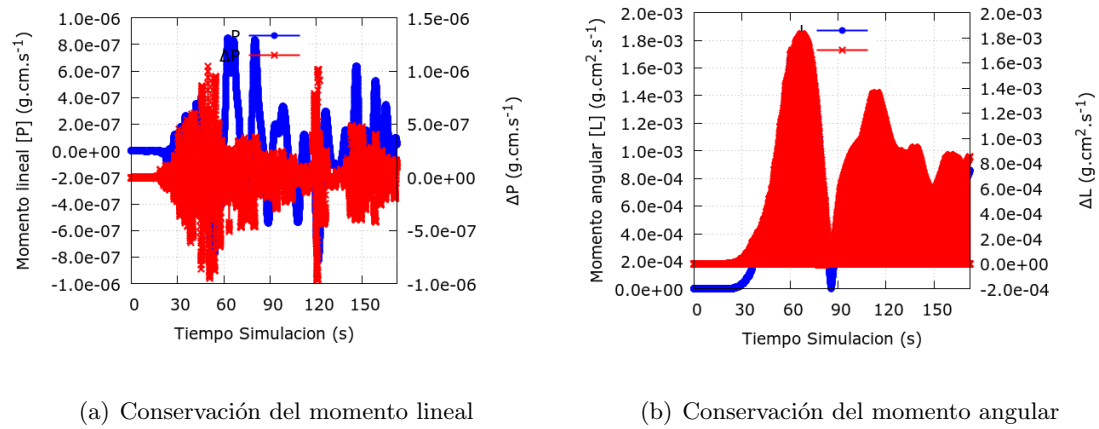


Figura 5.5: Conservación del momento lineal y angular en la simulación.

5.2. Equilibrio hidrostático de una estrella de tipo solar

Se genera un modelo de una estrella de la secuencia principal con masa solar a partir de un polítopo. El modelo se puede construir mediante la integración de la ecuación politrópica de *Lane-Endem* [21].

Este modelo se deja relajar mediante un código SPH explícito hasta que llega al equilibrio. La simulación con el código implícito comienza con este modelo y tiene como objetivo encontrar un nuevo punto de equilibrio en el código implícito con la gravedad activada.

Modelo inicial

Se dispone de un modelo en 3D con 8 000 partículas con simetría esférica y distribuidas de forma que reproduzcan el perfil de densidad politrópico. Se toma una masa similar a la del sol $M = 1M_{\odot}$ y se distribuye de tal modo que el radio máximo sea el solar $R = R_{\odot}$. que se corresponde con una densidad central de $\rho_c = 66 \text{ g.cm}^{-3}$. Como ecuación de estado se toma la de gas ideal con $\mu_i = 0.62$.

Resultados

El cuadro 5.2 describe el comportamiento de la ejecución del código.

Parts.	PC	$\tau_{Model} (s)$	$\tau_{CPU_{system}} (d)$	$\tau_{CPU_{user}} (d)$	Speedup	Iters.	Refs.
8 000	B	61 000	26.78	1.84	14.55	156	941

Cuadro 5.2: Tiempo de computación del código implícito. Los valores se muestran una vez el modelo consigue la estabilidad. La ejecución en el PC B se realiza con 16 threads en un servidor multinúcleo con 2 CPU Intel Xeon W5580 3.2Ghz y 48 GB DDR3-1333Mhz. El tiempo de computación, tanto de sistema como de usuario, se especifica en días. A la luz de estos números podemos deducir que si el sistema esta prácticamente en equilibrio, la matriz dispersa es muy homogénea en el valor de sus celdas y por tanto los preacondicionadores de PARDISO funcionan peor y prácticamente todo el tiempo de cálculo es utilizado en la inversión de la matriz.

En la figura 5.6 se muestran los tiempos de computación.

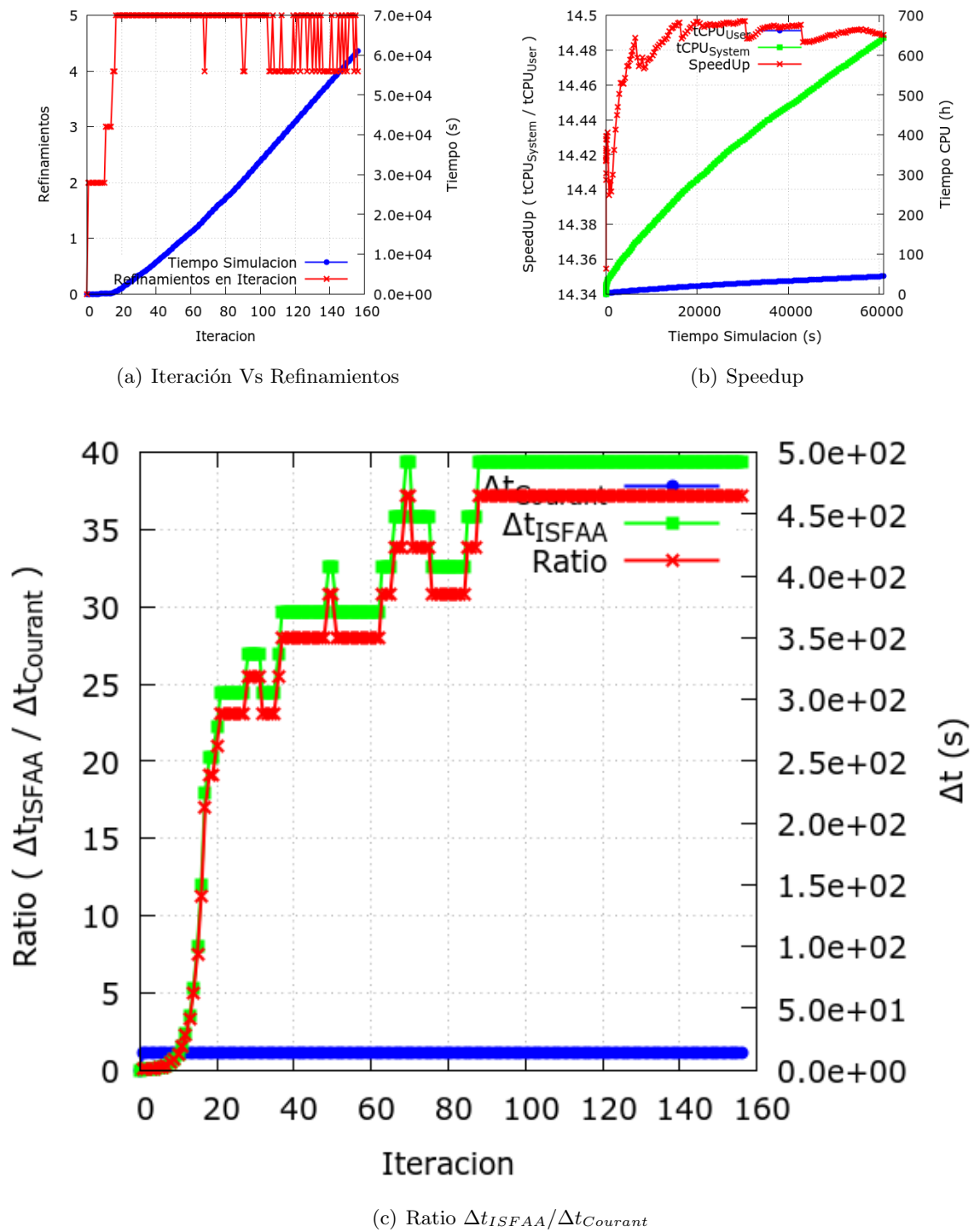
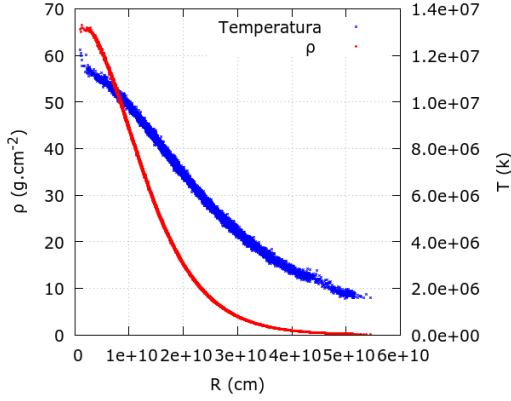
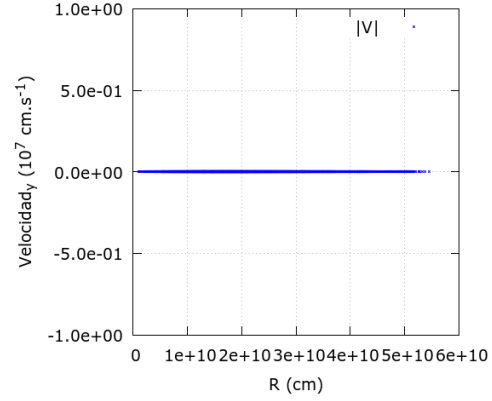


Figura 5.6: Tiempos de computación de la simulación con 8 000 partículas. En el panel 5.6(c) se puede observar que el ratio del paso de tiempo respecto del tiempo de Courant es 35 veces superior.

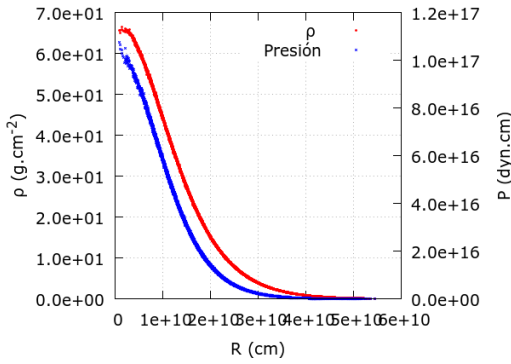
En la figura 5.7 se muestran los perfiles cuando el modelo llega a la estabilidad.



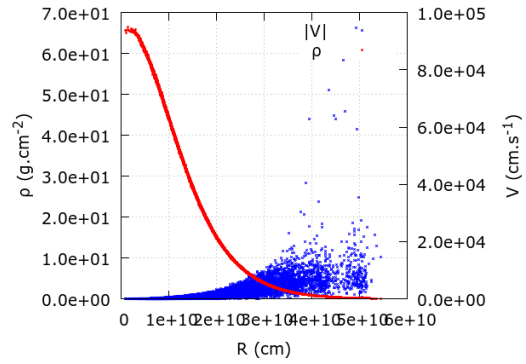
(a) Perfil inicial (ρ , T) en Iter=0



(b) Perfil inicial (V) en Iter=0



(c) Perfil final (ρ , P) en Iter=156



(d) Perfil final (ρ , V) en Iter=156

Figura 5.7: Perfiles de densidad, velocidad, presión y temperatura cuando la simulación consigue la estabilidad. En el panel 5.7(d) se observa que las velocidades de las partículas exteriores, aun siendo pequeñas, son las que determinan que la convergencia de la iteración. Es decir, los criterios estrictos pueden hacer que zonas sin interés para la simulación controlen el paso de tiempo de tiempo. Es tarea del científico tunear el código para establecer los criterios, en cada modelo, que eviten que las zonas no esenciales supongan un límite en la simulación.

En la figura 5.8 se puede observar la evolución temporal de las partículas del modelo.

En la figura 5.9 se puede observar la conservación de la energía del sistema.

En la figura 5.10 se puede observar la conservación del momento lineal y angular del sistema.

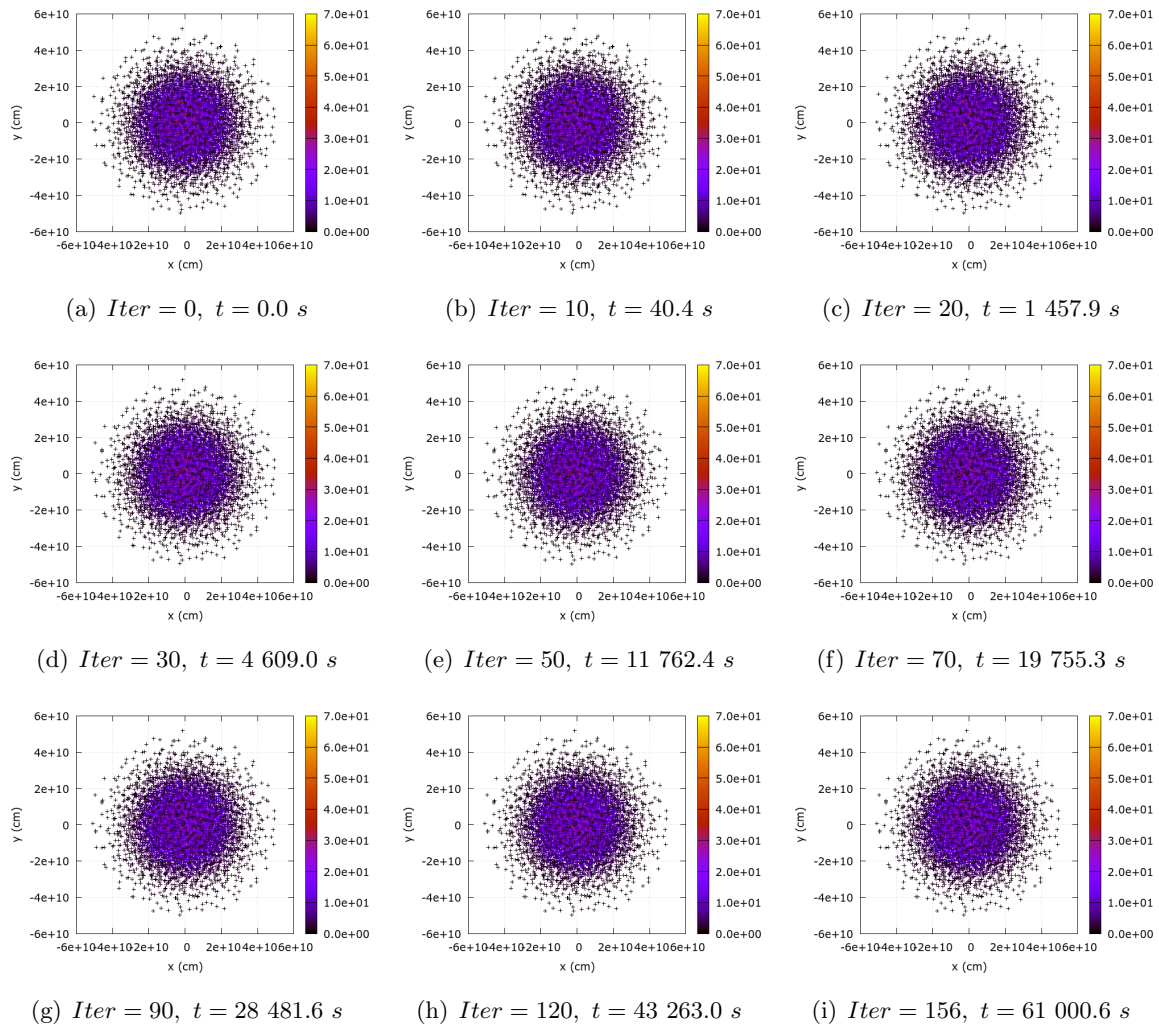


Figura 5.8: Evolución temporal de una estrella de tipo solar. Se muestra el mapa de color de la densidad (g/cm^3) en diferentes instantes de tiempo.

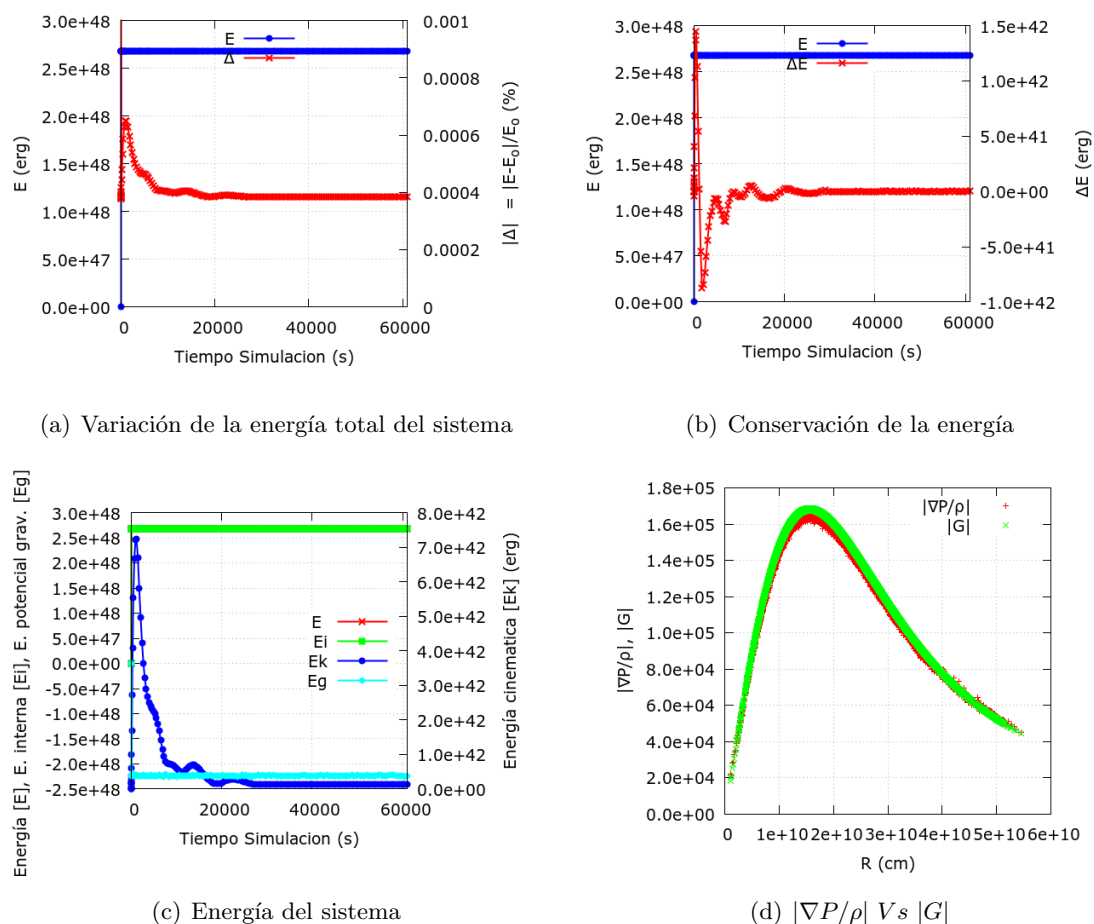


Figura 5.9: Conservación de la energía en la simulación y equilibrio del sistema. En el panel 5.4(d) se puede observar el ajuste con la gravedad en el equilibrio.

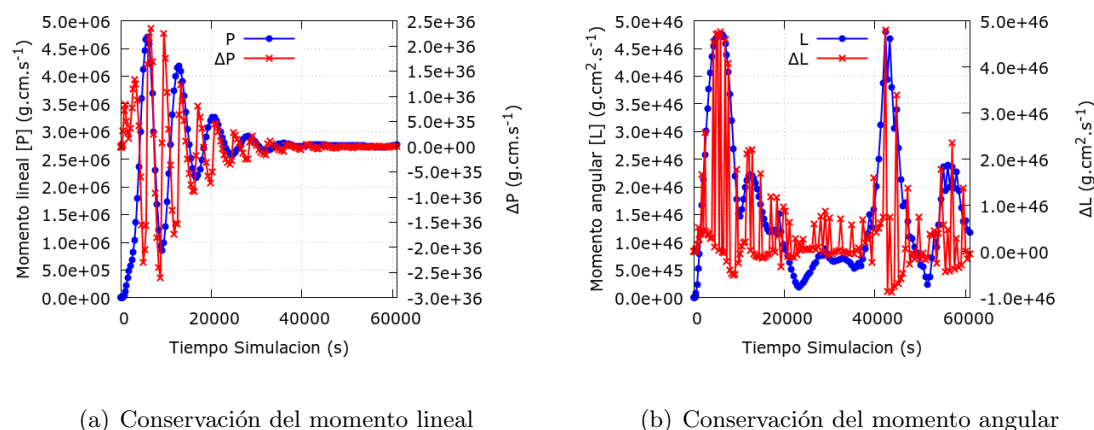


Figura 5.10: Conservación del momento lineal y angular en la simulación.

5.3. Equilibrio hidrostático de una enana blanca masiva ($m \simeq 1.15M_{\odot}$)

Una prueba más completa y de gran interés para caracterizar el comportamiento del código SPH implícito es la simulación de la hidrostática, por lo tanto independiente del tiempo, de la estructura de una enana blanca isoterma de $M_{wd} = 1.15 M_{\odot}$ con $N = 20\,000$ partículas.

Este modelo ha sido construido utilizando un SPH explícito con la misma física que luego se utilizará en el esquema implícito.

Es un objeto mucho más inestable que en el test del equilibrio hidrostático de una estrella de tipo solar.

Modelo inicial

Se dispone de un modelo inicial en 3D con 20 000 partículas y unos 100 vecinos reales.

Se aplican las siguientes personalizaciones del algoritmo:

- Gravedad implícita basada en el cálculo aproximado del potencial gravitatorio, según se ha descrito anteriormente.
- Ecuación de estado más sofisticada y realista, especialmente construida para su uso en objetos compactos y que incluye electrones degenerados, iones y radiación.
- Transporte de calor por conducción.

Además del caso que aquí se presenta, se han probado diferentes parámetros:

- Modelo inicial con diferente número de partículas, entre 5 000 y 40 000 partículas.
- Número de vecinos, entre 12 y 22 fraccionarios que equivalen a entre 50 y 100 reales.
- Gravedades diferentes: ley de gauss (considerando simetría esférica), directa explícita y implícita.
- Temperatura constante o temperatura variable.
- En caso de temperatura variable, con o sin conductividad térmica.

Resultados

En la figura 5.11 se muestra la distribución de la enana blanca en 3D con 20 000, proyectada en un plano, una vez que llega al equilibrio.

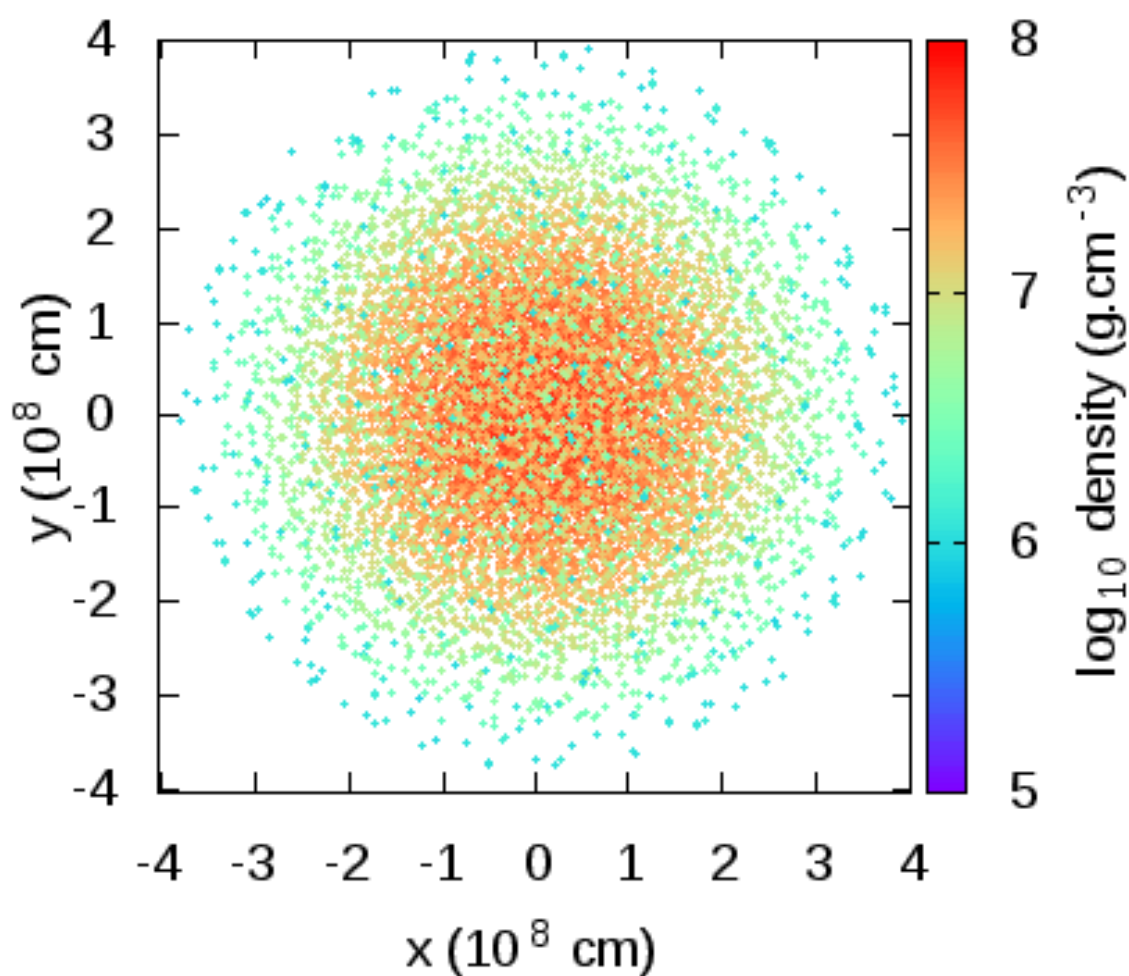


Figura 5.11: Evolución temporal a un tiempo $t = 25$ s. Se muestra el mapa de color de la densidad ($g.cm^{-3}$).

En la figura 5.12 se muestra el perfil del gradiente de presión dividido para la densidad

$|\nabla P/\rho|$ comparado con el valor absoluto de la gravedad, a un tiempo de $t = 25$ s.

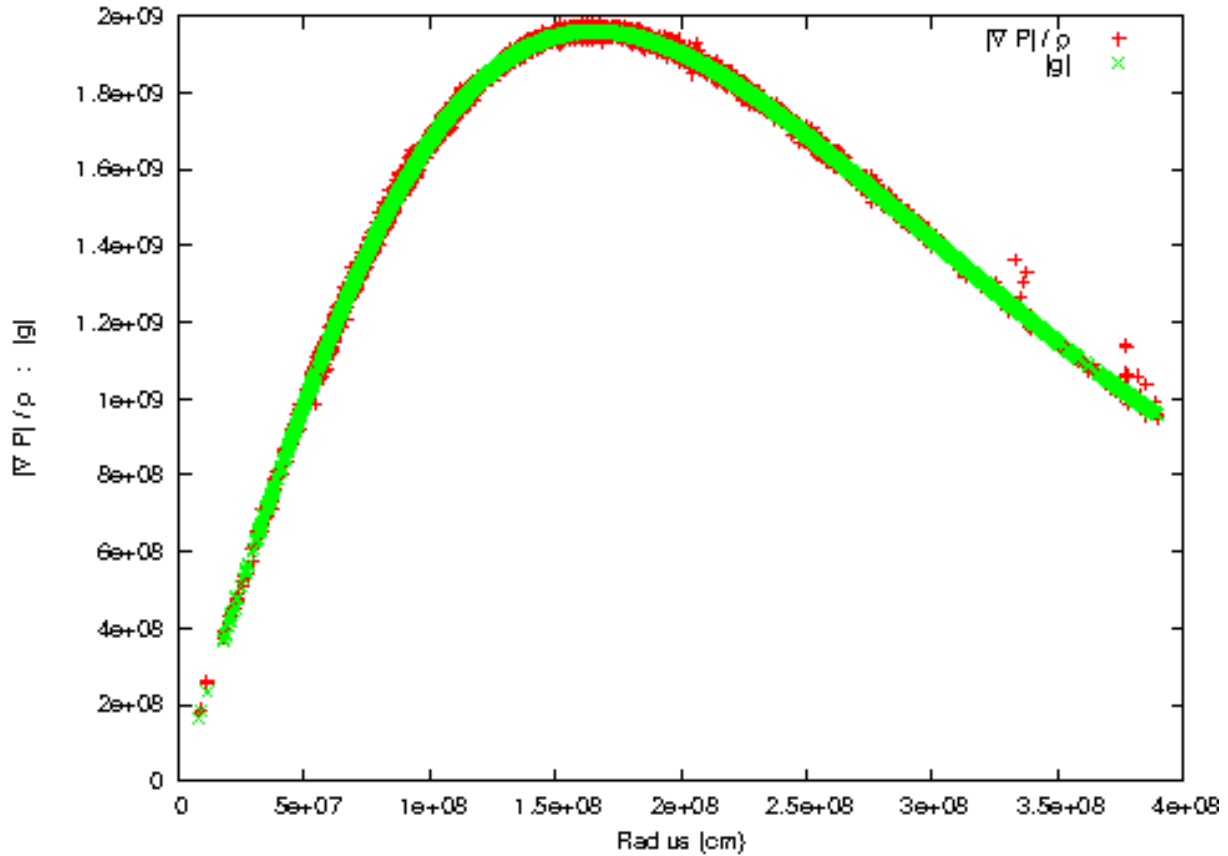


Figura 5.12: Perfil del gradiente de presión dividido para la densidad comparado con el valor absoluto de la gravedad, a $t = 25$ s. La correspondencia entre ambos perfiles es excelente y la enana blanca alcanza un buen equilibrio mecánico.

La figuras 5.13 y 5.14 se muestra la energía y su evolución en la simulación. Son los estados finales de la enana blanca durante la evolución hacia la estabilidad a $t = 25$ s.

Vemos que la energía cinética decae rápidamente cuando se acerca al equilibrio mecánico y térmico. Después de 25 segundos la energía cinética se ha reducido muchos ordenes de magnitud en comparación con la energía de la estrella.

Sin embargo, hay todavía un nivel de ruido que impide un crecimiento adecuado del paso de tiempo. El paso de tiempo siempre se mantuvo por debajo de las 50 veces el tiempo de Courant, similar al caso anterior, pero todavía insuficiente para llevar a cabo estudios prácticos sobre la evolución cuasi-hidrostática de las estructuras estelares.

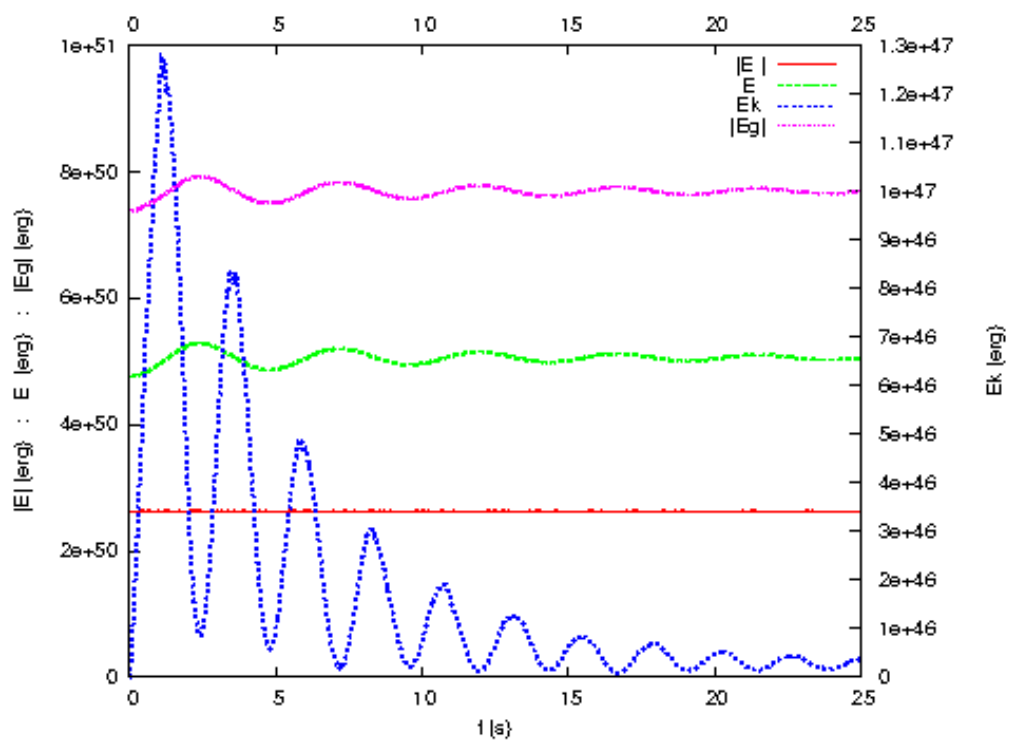


Figura 5.13: Energía del sistema.

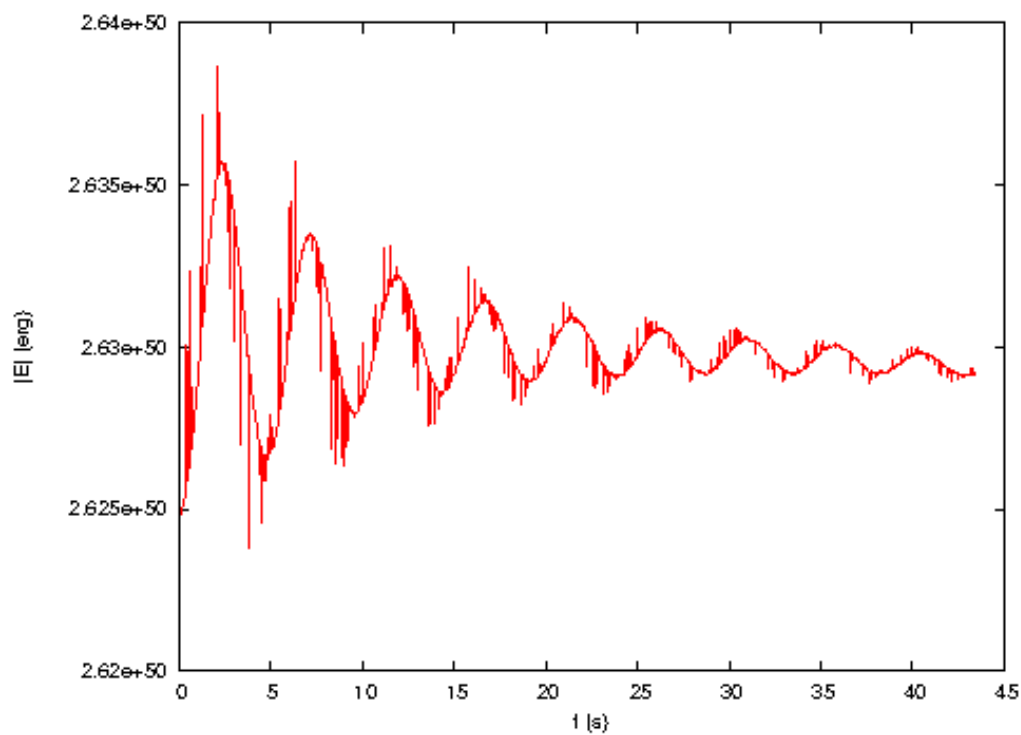


Figura 5.14: Variación de la energía.

5.4. Aproximación implícita cuasí-hidrostática

En este tipo de modelos las velocidades no entran explícitamente en el esquema de cálculo y el tiempo de evolución está determinado por el proceso físico, no por el tiempo de Courant.

El equilibrio de las ecuaciones se consigue mediante la ecuación del momento modificada (el término de la velocidad es nulo) y la ecuación de la energía modificada, según vemos a continuación:

$$\mathbf{a} = 0 = - \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \nabla_i \mathbf{W}_{ij}(h_i) + \frac{P_j}{\Omega_j \rho_j^2} \nabla_i \mathbf{W}_{ij}(h_j) \right) \quad (5.3)$$

$$\dot{U} = \frac{d U_i}{d t} = \frac{P_i}{\Omega_i \rho_i^2} \cdot (\rho_i - \rho_i^{(n-1)}) \quad (5.4)$$

$$\rho = \sum_{j=1}^N m_j W_{ij}(h_i) \quad (5.5)$$

$$h = \left(\frac{K}{\rho} \right)^{1/d} \quad (5.6)$$

Cuadro 5.3: Discretización de las ecuaciones del SPH: Esquema cuasí-hidrostático

Modelo inicial

Para modelo el inicial tomaremos una distribución 1D de 360 partículas colocadas de forma circular en 2D (es decir un sistema 1D cerrado sobre sí mismo), de tal modo que no tengamos problemas con los contornos. Se toman 18 vecinos reales. Un dato curioso es que en este modelo 1D circular la Ω tiene un valor de 0.5.

Inicialmente se realizará una relajación del modelo, encontrando un equilibrio en una única iteración. Para ello relajaremos el criterio de refinamientos máximos, colocándolo en 50 en vez de los 5 del resto de los tests. En esta primera fase no se activa la ecuación de la energía, es decir con el paso de los refinamientos la energía permanece constante.

Una vez alcanzado el equilibrio se inyectará energía de forma explícita, en función del paso de tiempo entre iteraciones, de forma que el modelo tenderá a expansionarse sin que la velocidad se encuentre explícitamente en el esquema. Para este caso se activa la ecuación de la energía en el esquema, para que distribuya la energía inyectada entre las partículas.

Para la inyección de energía se toma un paso de tiempo fijo de $\Delta t = 10^2$ s.

Resultados

El modelo consigue la estabilidad y posteriormente es capaz de expansionarse durante varias iteraciones.

Una vez que se empieza a inyectar energía el sistema se expande, pero a pesar de ello, con el paso de las iteraciones aparecen pequeñas diferencias de simetría en el modelo, que provocan acumulaciones de densidad en la línea circular de partículas. Estas diferencias acaban afectando a la totalidad de las partículas del modelo y provocan una pérdida de simetría global. De todos modos, el objetivo está cumplido, durante varias iteraciones el modelo logra expansionarse debido a la inyección de energía.

Con este test se demuestra que es posible trabajar con este tipo de modelos con un código implícito y es un buen punto de partida para poder explorar las posibilidades de los sistemas cuasi-hidrostáticos. Este tipo de sistemas no son objetivo de los códigos explícitos por la limitación en el paso de tiempo de Courant.

En la figura 5.15 se muestran los tiempos de computación.

En la figura 5.16 se puede observar la evolución temporal de las partículas del modelo.

En la figura 5.17 se puede observar la conservación de la energía y del momento lineal.

En la figura 5.18 se muestran los perfiles cuando el modelo llega a Iter=14

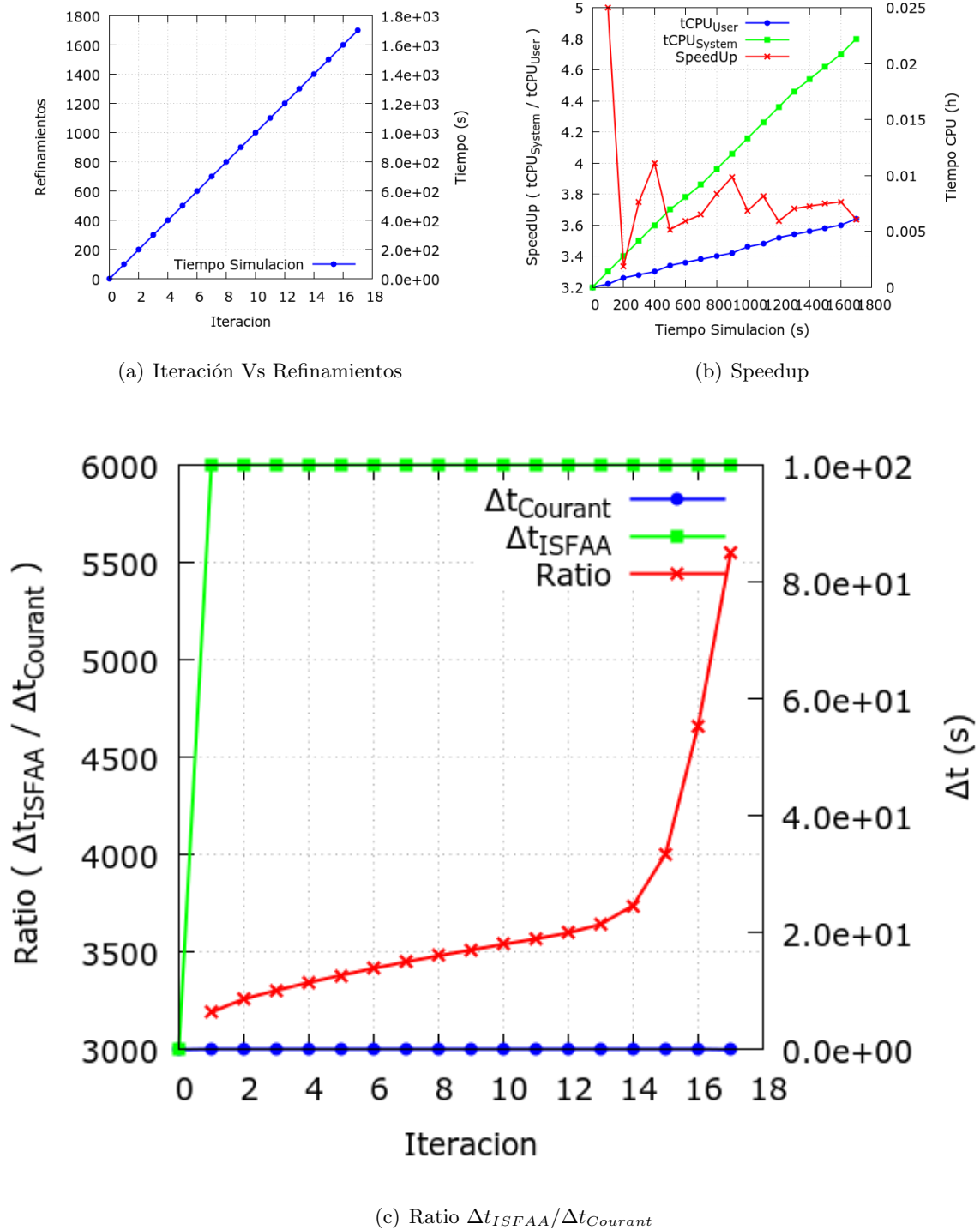


Figura 5.15: Tiempos de computación de la simulación con 360 partículas. En el panel 5.15(c) se puede observar que el ratio del paso de tiempo respecto del tiempo de Courant es enorme para este tipo de modelos. Este ratio depende del paso de tiempo escogido, si escogemos un paso de tiempo varios órdenes de magnitud mayor, veremos un ratio también varios ordenes de magnitud mayor.

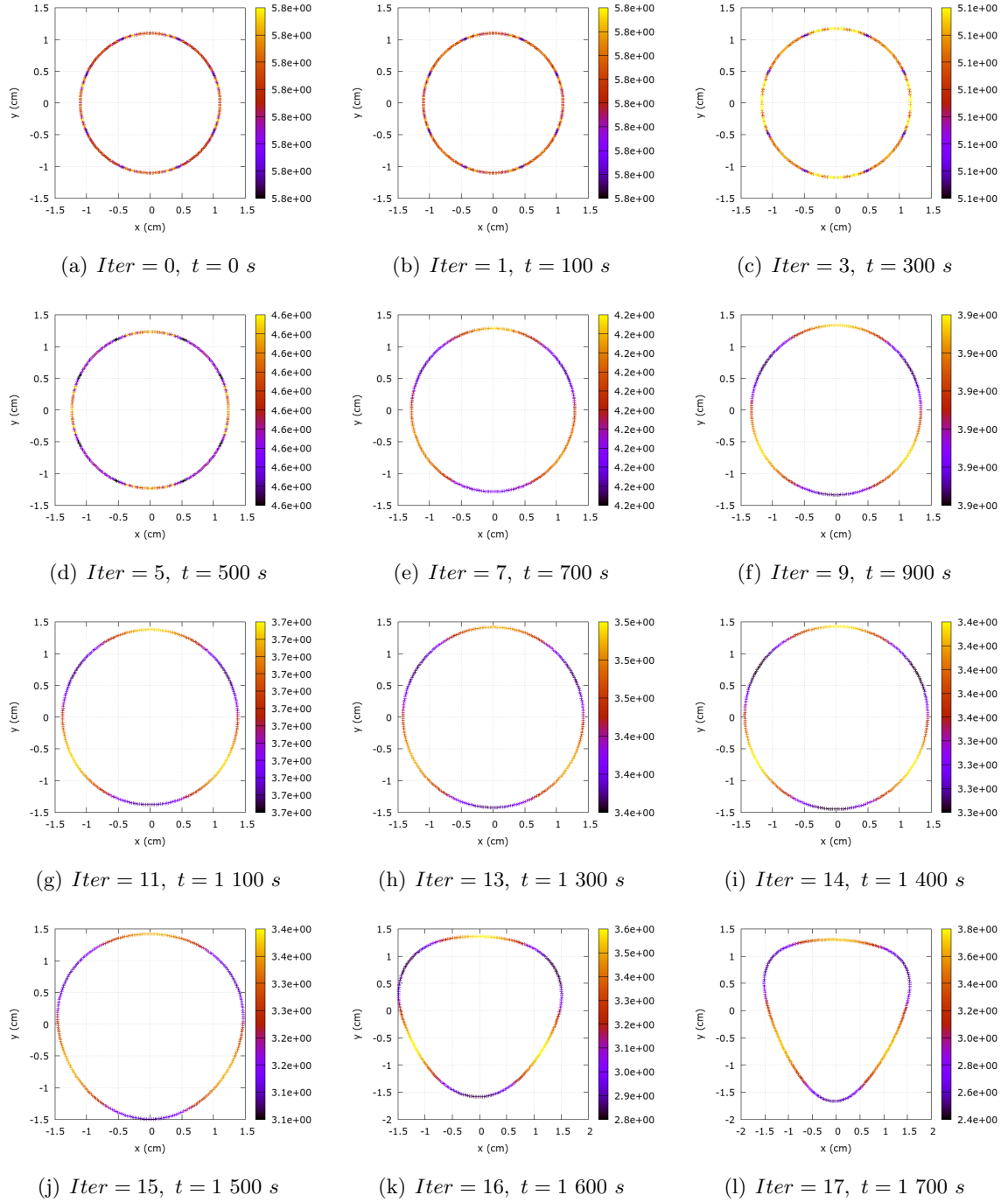
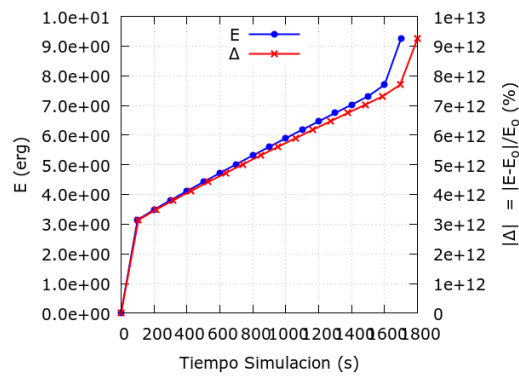
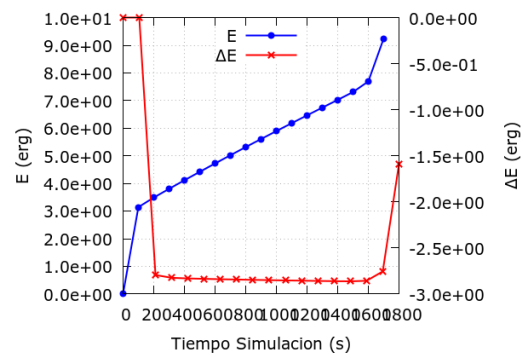


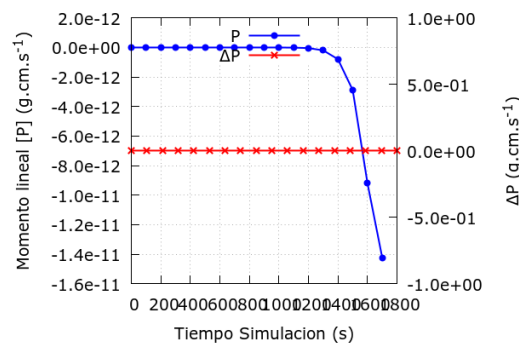
Figura 5.16: Evolución temporal del sistema cuasi-hidroestático. Se muestra el mapa de color de la densidad en diferentes instantes de tiempo. Se observa que en los paneles 5.16(j), 5.16(k) y 5.16(l) el modelo ya se encuentra totalmente desequilibrado.



(a) Variación de la energía total del sistema



(b) Conservación de la energía



(c) Conservación del momento lineal

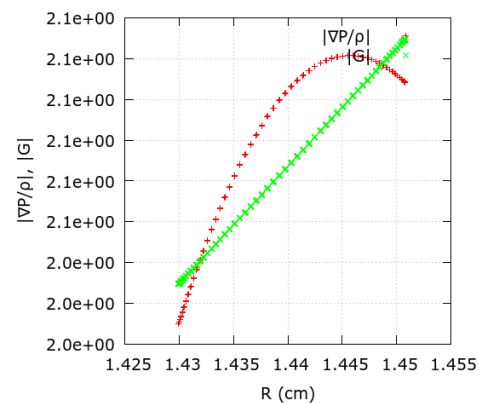
(d) $|\nabla P/\rho|$ Vs $|G|$

Figura 5.17: Conservación de la energía y el momento lineal en la simulación. En el panel 5.4(d) se puede observar el ajuste con la gravedad en la Iter=14, se ve claro que deberían coincidir pero a pesar de que el modelo mantiene la estructura. Aquí comienza el desequilibrio que acaba con el sistema.

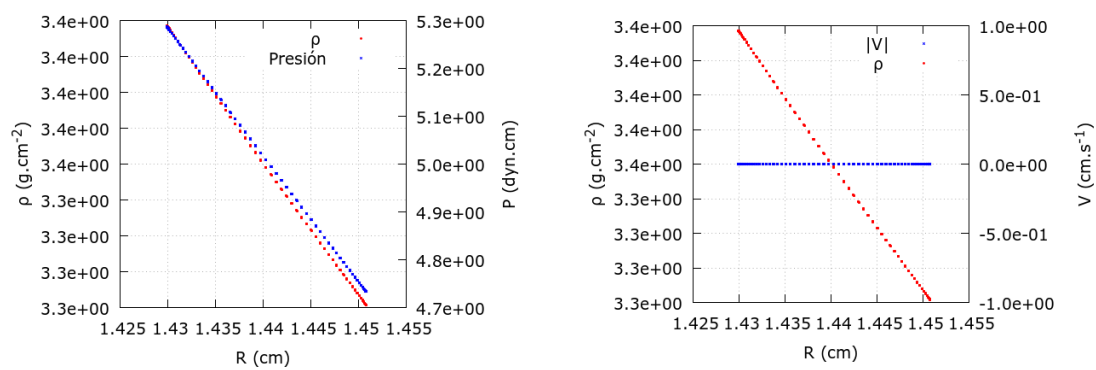
(a) Perfil final (ρ , P) en Iter=156(b) Perfil final (ρ , V) en Iter=156

Figura 5.18: Perfiles de densidad, presión y velocidad a Iter=14. En el panel 5.18(a) se observan las pequeñas diferencias de densidad que provocan la inestabilidad. En el panel 5.18(b) se observa que las velocidades del modelo son cero.

Capítulo 6

Conclusiones

En esta tesis se ha construido un código implícito de SPH (denominado ISFAA) cuyo objetivo final sería estudiar la evolución cuasi-hidrostática en condiciones de interés para la astrofísica. El formalismo matemático utilizado es el del estándar SPH, basado en el principio variacional e implementa viscosidad artificial, gravedad y transferencia de calor por conducción. El procedimiento para resolver las ecuaciones de Euler, sin embargo, es muy diferente ya que las ecuaciones se aproximan de manera implícita. A partir de una solución aproximada que se va refinando de forma iterativa mediante un Newton-Raphson multidimensional hasta que la convergencia hacia la solución correcta se consigue, con una diferencia muy pequeña respecto del valor real.

En su estado actual, ISFAA es capaz de manejar unas 30 000 partículas en 2D y unas 10 000 partículas en 3D en un PC estándar con un Intel Core i7 y 16GB de RAM. Los resultados obtenidos por ISFAA mediante la ejecución de tests (Explosión puntual, inestabilidades, etc.) concuerdan con los resultados del SPH explícito. Sin embargo, el verdadero poder del método implícito solo se demuestra en situaciones cuasi-hidrostáticas donde los pasos de tiempo deben de ser muy largos por que el sistema evoluciona muy lentamente. En este sentido, se ha aplicado con éxito el método implícito, como en la simulación de la estabilización de un polítropo de masa solar o una enana blanca masiva. Con del código implícito actual se consiguen ratios de aumento del paso de tiempo de 100 veces respecto del tiempo de Courant, τ_c , en los modelos casi-hidrostáticos. También se ha experimentado con modelos muy básicos sin velocidad que permitirían desarrollar el potencial de un código implícito, con resultados esperanzadores de varios órdenes de magnitud superior al τ_c .

6.1. Objetivos conseguidos

Se pueden describir los objetivos conseguidos en la tesis en los siguientes puntos:

- Es factible construir un código SPH implícito (como ISFAA) que incorpore las características más modernas de la técnica SPH, según el principio variacional, con una inversión computacional aceptable.
- ISFAA incorpora muchas de las piezas de física necesarias para realizar simulaciones de interés astrofísico: viscosidad artificial para el tratamiento de choques, diferentes implementaciones de la gravedad, transferencia de calor por conducción, etc.
- Los test realizados demuestran que con ISFAA pueden realizarse simulaciones tomando pasos de tiempo superiores al tiempo de Courant (τ_c) sin afectar a la estabilidad. Parece que con la formulación utilizada hay una limitación en el paso de tiempo máximo que puede tomarse sin que peligre la estabilidad del modelo. En particular es posible utilizar pasos de tiempo cercanos a $100 \tau_c$, dependiendo del modelo a tratar. Este valor está de acuerdo con los resultados de otros autores que han investigado sobre códigos pseudo-implícitos como *Almgren y colaboradores, 2006-2008* [1] [2] [3].
- En consecuencia, los códigos SPH implícitos todavía no son competitivos, en modelos hidrodinámicos, respecto a los explícitos debido a la diferencia de consumo computacional inherente al método implícito. Sería necesario un ratio en el paso de tiempo implícito de $10^{4-6} \tau_c$, en vez del $10^2 \tau_c$ actual. Como se ha observado en capítulos anteriores, esto puede ser factible formulando de entrada un modelo hidrostático (es decir sin velocidad explícita en las ecuaciones).
- Siendo la gravitación un elemento fundamental de la mayoría de cálculos astrofísicos, se ha realizado una comparación de diferentes implementaciones del cálculo de esa fuerza. La conclusión es que el mejor algoritmo es el que utiliza el potencial gravitatorio calculado por interacción directa para calcular la fuerza. Esto es debido a que al calcularse la fuerza a través de un gradiente y siendo el cálculo del gradiente una propiedad local de la técnica SPH, implica que la matriz del sistema sigue siendo altamente dispersa.
- La limitación en el paso de tiempo se mejora utilizando códigos implícitos de tipo hidrostático, es decir, sin términos de velocidad. Donde, la solución está regida por la variaciones en la entropía del sistema debido a agentes externos (reacciones nucleares, acreción, etc.). En la última aplicación se experimenta en este sentido, aunque todavía es necesaria más investigación para obtener resultados realmente rendibles.

6.2. Avances en el método implícito

Se han conseguido mejorar en los siguientes apartados, respecto del trabajo previo de *Knapp, 2000* [47]:

- SPH Variacional: Implementa la formulación lagrangiana del SPH que se usa hoy en día.
- Como consecuencia natural del SPH Variacional integra la variación de la distancia de interpolación como variable independiente en el esquema de Newton-Raphson.
- Gravedad: Se han implementado varias técnicas para el cálculo de la gravedad, prestando especial atención a una técnica novedosa para tratar la gravedad, de forma implícita, mediante el potencial gravitatorio.
- Kernel: El código ISFAA implemente el kernel Sinc que con una formulación tensorial y frente al *Cubic spline*, mejora la precisión de la interpolación y elimina el “pairing instability” pudiendo variar el parámetro n y ajustarlo según sea necesario.
- Reducción de las variables del problema: Se ha establecido un método fiable para reducir la ecuación de la densidad y la velocidad integrándolas dentro de la ecuación del momento de forma implícita. Esto obtiene una gran ventaja al reducir de forma muy acusada el número de casillas de la matriz dispersa y, por tanto, reduciendo el tiempo de computación.
- Paralelización del método: Se ha paralelizado el método mediante OpenMP y se han utilizado bibliotecas algorítmicas específicas para el tratamiento de sistemas de ecuaciones (finalmente se ha optado por PARDISO).
- Parametrización y modularidad: Se ha escrito el código de tal modo que es multidimensional y parametrizable de modo que se puede utilizar para multitud de problemas usando un único código base. Este código puede ser fácilmente modificable y debugable gracias a la modularidad del diseño.

6.3. Viabilidad del método implícito

La limitación del método implícito es el tiempo de computación debido a la inversión de la matriz, que tiene un coste computacional superior al resto de cálculos del programa (entre cuadrático y cúbico) y por tanto es un factor limitante. Un aumento en el número de ecuaciones se traduce en un mayor número de celdas no vacías de la matriz y que la computación aumente en un factor similar al coste computacional de la matriz, $\Theta(n^i)$, siendo n el tamaño de la matriz

cuadrada y i un valor entre 2 – 3, dependiendo del problema tratado, el número de partículas y el método utilizado en la resolución de la inversión de la matriz.

La memoria también es un factor limitante, pero en menor medida que la computación. Se ha comprobado que si aumentamos el número de partículas al doble (a partir de una cantidad mínima de unas 5.000 partículas) el tamaño ocupado en memoria es algo aproximadamente el doble, es decir sigue un coste lineal $\Theta(n)$ según el número de partículas.

En función del número de partículas la memoria sigue un aumento lineal en tamaño, por un aumento entre cuadrático y cúbico en coste computacional.

Estos factores provocan que el número de partículas que se pueden simular es varias veces inferior al del método explícito. Estos efectos han sido paliados parcialmente con la reducción de variables pero todavía sigue siendo el factor que marca el coste de computación y el consumo de memoria. Otro factor importante es la precisión de la interpolación, determinada con el número de vecinos ideal, cuanto más exigente seamos con la precisión mayor tendrá que ser el número de vecinos implicados en el cálculo y por tanto mayor será la tasa de relleno de la matriz y menor será la compactación de la misma.

Por todo ello, la aplicación realmente usable para el método implícito es en los modelos cuasi-hidrostáticos que no pueden ser calculados por el método explícito y si por el implícito, frente a los modelos hidrodinámicos que ocurre lo contrario.

6.4. Trabajo futuro

Varias mejoras de ISFAA se están valorando para poder aplicarlo a escenarios astrofísicos evolutivos. Estas y otras mejoras generales se describen a continuación:

- Seguir investigando sobre los modelos sin velocidad que pueden ser una gran ventaja para un código implícito SPH como ISFAA.
- Desarrollar una versión 2D axisimétrica de ISFAA, basada en los trabajos previos de *García-Senz y colaboradores, 2009* [27] y *Relaño, 2012* [73]. Los códigos Axisimétricos aprovechan la simetría axial que se puede observar en muchos procesos de la naturaleza. Por lo tanto, estos códigos son capaces de describir fenómenos 3D que trabajan en cualquier plano que contiene el eje de simetría. Esto permite simular un modelo 3D utilizando un código hidrodinámico 2D con la consiguiente mejora en la resolución, circunstancia ideal que permitiría paliar los problemas de resolución del código ISFAA. Actualmente hay una gran cantidad de códigos eulerianos que aprovechan esta circunstancia, muchos de ellos de

tipo AMR (como por ejemplo *Ogando y Velarde, 2001* [68] o *Zhang y MacFadyen, 2006* [105]). Comparativamente el número de códigos SPH que hacen uso de la aproximación axisimétrica es mucho menor.

- Integrar la formulación IAD del SPH que, aparentemente, mejorará la resolución de los cálculos a mismo número de partículas y simplificará una parte de las derivadas que se incluyen en la matriz (*García-Senz y colaboradores, 2012* [26], [16]).
- Aplicar el algoritmo *z-orden* del método de Morton (*Morton, 1966* [64]) para ordenar la matriz y mejorar el precondicionamiento de la misma antes de su inversión. Este algoritmo esta ya implementando en el código, pero todavía no se han construido tests para poder realizar pruebas de tiempo que observen su impacto, Una vez comprobada su eficiencia puede ser utilizado en cualquier modelo que se simule. La ordenación z-order permite aumentar la compactación de la matriz, reorganizando el índice de las partículas cercanas espacialmente, acercándolas a la diagonal de la matriz y reduciendo de este modo el tiempo de cálculo.
- Estudiar formas de estabilizar mejor el esquema numérico, para poder trabajar con ratios de pasos de tiempo, respecto del τ_c , más grandes que los actuales.
- Añadir reacciones nucleares mediante una red nuclear.
- Migrar de PARDISO 5.0 a la implementación de Intel MKL PARDISO que se puede encontrar y usar de forma libre y gratuita en la versión de 2016.

Además de los anteriores, otro de los objetivos principales es migrar la paralelización del código, actualmente en CPUs, a GPUs (mediante la recodificación en el lenguaje de programación CUDA) y a procesadores Intel Xeon Phi (véase apéndice G). La paralelización en GPUs o en Xeon Phi hoy en día es una realidad y esta siendo incorporada a todos los códigos SPH actuales. Este es un requisito necesario para que puedan correr en los nuevos supercomputadores, los códigos no adaptados no serán competitivos frente a los que sí utilicen estas tecnologías, y por tanto, puedan aprovechar todo el potencial de cálculo disponible.

Parte II

Apéndices

Apéndice A

Kernel

El valor del kernel es diferente en función del parámetro $v = |r_{ij}|/h$. Este parámetro muestra la relación entre la longitud de suavizado h de la partícula y la distancia a la partícula j (r_{ij}). Para kernels con soporte compacto habitualmente la distancia que se toma es $|r_{ij}| < 2h$. Las derivadas del kernel a partir de distancias de $|r_{ij}| \geq 2h$, tanto el valor como las derivadas son igual a cero. Las siguientes secciones muestran las derivadas de cada una de las partes que intervienen en las derivadas del kernel.

En adelante, y por claridad en las ecuaciones, se toman variables genéricas (como X , H , D , *etc.*) que pueden ser sustituidas por variables concretas (como x_i , z_j , h_j , *etc.*). Para obtener todas las ecuaciones reales hay que realizar todas las combinaciones de variables. En la leyenda de cada ecuación se especifican las opciones disponibles.

Upsilon ($v = |r_{ij}|/h$)

$\frac{\partial v}{\partial X} =$	$\frac{1}{h} \frac{\partial r_{ij}}{\partial X}$	$\frac{\partial v}{\partial H} =$	$-\frac{1}{h} v \frac{\partial h}{\partial H}$
$\frac{\partial^2 v}{\partial X_2 \partial X} =$	$\frac{1}{h} \frac{\partial^2 r_{ij}}{\partial X_2 \partial X}$	$\frac{\partial^2 v}{\partial X_2 \partial H} =$	$-\frac{1}{h} \frac{\partial v}{\partial X_2} \frac{\partial h}{\partial H}$
$\frac{\partial^2 v}{\partial H_2 \partial X} =$	$-\frac{1}{h^2} \frac{\partial r_{ij}}{\partial X} \frac{\partial h}{\partial H_2}$	$\frac{\partial^2 v}{\partial H_2 \partial H} =$	$\left[\frac{1}{h^2} v \frac{\partial h}{\partial H_2} - \frac{1}{h} \frac{\partial v}{\partial H_2} \right] \frac{\partial h}{\partial H}$

Cuadro A.1: Kernel Sinc: Derivadas del parámetro v . Cambiar variables genéricas X y X_2 por: x_i, x_j ; y_i, y_j ; z_i, z_j . Cambiar variables genéricas H y H_2 por: h_i, h_j .

Distancia partícula $i - j$ (r_{ij})

▪ $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j| = [(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{1/2}$, distancia entre i y j .

$\frac{\partial r_{ij}}{\partial x_i} =$		$-\frac{\partial r_{ij}}{\partial x_j} =$		$\frac{(x_i - x_j)}{r_{ij}}$
$\frac{\partial^2 r_{ij}}{\partial x_i \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_j \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_i \partial x_j} =$	$\frac{\partial^2 r_{ij}}{\partial x_j \partial x_j} =$	$\frac{r_{ij}^2 - (x_i - x_j)^2}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial y_i \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_j \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_i \partial x_j} =$	$\frac{\partial^2 r_{ij}}{\partial y_j \partial x_j} =$	$-\frac{(x_i - x_j)(y_i - y_j)}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial z_i \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_j \partial x_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_i \partial x_j} =$	$\frac{\partial^2 r_{ij}}{\partial z_j \partial x_j} =$	$-\frac{(x_i - x_j)(z_i - z_j)}{r_{ij}^3}$
$\frac{\partial r_{ij}}{\partial y_i} =$		$-\frac{\partial r_{ij}}{\partial y_j} =$		$\frac{(y_i - y_j)}{r_{ij}}$
$\frac{\partial^2 r_{ij}}{\partial x_i \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_j \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_i \partial y_j} =$	$\frac{\partial^2 r_{ij}}{\partial x_j \partial y_j} =$	$-\frac{(x_i - x_j)(y_i - y_j)}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial y_i \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_j \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_i \partial y_j} =$	$\frac{\partial^2 r_{ij}}{\partial y_j \partial y_j} =$	$\frac{r_{ij}^2 - (y_i - y_j)^2}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial z_i \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_j \partial y_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_i \partial y_j} =$	$\frac{\partial^2 r_{ij}}{\partial z_j \partial y_j} =$	$-\frac{(y_i - y_j)(z_i - z_j)}{r_{ij}^3}$
$\frac{\partial r_{ij}}{\partial z_i} =$		$-\frac{\partial r_{ij}}{\partial z_j} =$		$\frac{(z_i - z_j)}{r_{ij}}$
$\frac{\partial^2 r_{ij}}{\partial x_i \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_j \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial x_i \partial z_j} =$	$\frac{\partial^2 r_{ij}}{\partial x_j \partial z_j} =$	$-\frac{(x_i - x_j)(z_i - z_j)}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial y_i \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_j \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial y_i \partial z_j} =$	$\frac{\partial^2 r_{ij}}{\partial y_j \partial z_j} =$	$-\frac{(y_i - y_j)(z_i - z_j)}{r_{ij}^3}$
$\frac{\partial^2 r_{ij}}{\partial z_i \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_j \partial z_i} =$	$-\frac{\partial^2 r_{ij}}{\partial z_i \partial z_j} =$	$\frac{\partial^2 r_{ij}}{\partial z_j \partial z_j} =$	$\frac{r_{ij}^2 - (z_i - z_j)^2}{r_{ij}^3}$

Cuadro A.2: Kernel Sinc: Derivadas de la distancia \mathbf{r}_{ij} .

Parámetro Ω_i (gradiente de h_i)

En el formalismo *Euler-Lagrange* (Sec. 2.2.3) identificamos el parámetro Ω . (ecuación A.3).

	$\rho_i = K h_i^{-d} \quad ; \quad K = \eta_{h_i}^d m_i \quad ; \quad h_i = \eta \left(\frac{m_i}{\rho_i} \right)^{1/d} \quad ; \quad \frac{\partial h_i}{\partial \rho_i} = -\frac{1}{d} \frac{h_i}{\rho_i}$
$\Omega_i =$	$1 - \frac{\partial h_i}{\partial \rho_i} \sum_{j=1}^N m_j \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial h_i} = 1 + \frac{1}{d} \frac{h_i}{\rho_i} \sum_{j=1}^N m_j \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial h_i}$
$\frac{\partial \Omega_i}{\partial X} =$	$\frac{1}{d} \frac{h_i}{\rho_i} \sum_{j=1}^N m_j \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial X \partial h_i}$
$\frac{\partial \Omega_i}{\partial \rho} =$	$-\frac{1}{d} \frac{h_i}{\rho_i^2} \frac{\partial \rho_i}{\partial \rho} \sum_{j=1}^N m_j \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial h_i}$
$\frac{\partial \Omega_i}{\partial h} =$	$\frac{1}{d} \frac{1}{\rho_i} \sum_{j=1}^N m_j \left[\frac{\partial h_i}{\partial h} \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial h_i} + h_i \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial h \partial h_i} \right]$

Cuadro A.3: Kernel Cubic Spline: Parámetro Ω . Cambiar variable genérica X por: $x_i, x_j; y_i, y_j; z_i, z_j$.

Signos en las derivadas, según los índices del Kernel W_{ij}

Dependiendo del orden de los índices mudos i, j el valor del kernel y sus derivadas varían. El cuadro A.4 resume la relación entre índices. $\mathbf{W}_{ij}(h_i) = \mathbf{W}_{ij}(h_i)(|\mathbf{r}_i - \mathbf{r}_j|, h_i) = \mathbf{W}_{ji}(h_i)$

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial x_i} = -\frac{\partial \mathbf{W}_{ij}(h_i)}{\partial x_j} \\ \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial y_i} = -\frac{\partial \mathbf{W}_{ij}(h_i)}{\partial y_j} \\ \frac{\partial \mathbf{W}_{ij}(h_i)}{\partial z_i} = -\frac{\partial \mathbf{W}_{ij}(h_i)}{\partial z_j} \end{array} \right.$$

$$\left\{ \begin{array}{l} \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial x_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial x_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial x_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial x_i} \\ \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_i \partial y_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_j \partial y_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_i \partial y_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_j \partial y_i} \\ \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial z_i \partial z_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial z_j \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial z_i \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial z_j \partial z_i} \end{array} \right. \quad \left\{ \begin{array}{l} \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial y_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial y_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial y_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial y_i} \\ \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial z_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_i \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial x_j \partial z_i} \\ \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_i \partial z_i} = \frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_j \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_i \partial z_j} = -\frac{\partial^2 \mathbf{W}_{ij}(h_i)}{\partial y_j \partial z_i} \end{array} \right.$$

Cuadro A.4: Kernel Cubic Spline: Signos dependiendo del orden de índices i, j

A.1. Kernel Cubic spline: $W(v, h)$

$$W(v, h) = C \cdot W'(v, h) = \frac{1}{K h^d} \cdot \begin{cases} W'_I(v, h) = 1 - \frac{3}{2}v^2 + \frac{3}{4}v^3, & (0 < v < 1) \\ W'_{II}(v, h) = \frac{1}{4}(2 - v)^3, & (1 \leq v < 2) \\ 0, & (v \leq 0 \wedge v \geq 2) \end{cases} \quad (\text{A.1})$$

$$\blacksquare C = \frac{1}{K h^d}, \quad \text{según dimensiones: } \left[\left(d = 1, K = \frac{2}{3} \right), \left(d = 2, K = \frac{10}{7 \cdot \pi} \right), \left(d = 3, K = \frac{1}{\pi} \right) \right].$$

Derivadas de $W(v, h)$

$\frac{\partial [\mathbf{W}(v, h)]}{\partial D} =$	$\frac{\partial C}{\partial D} \cdot W'(v, h) + C \frac{\partial W'(v, h)}{\partial D}$
$\frac{\partial^2 [\mathbf{W}(v, h)]}{\partial D_2 \partial D} =$	$\frac{\partial^2 C}{\partial D_2 \partial D} \cdot W'(v, h) + \frac{\partial C}{\partial D} \cdot \frac{\partial W'(v, h)}{\partial D_2} + \frac{\partial C}{\partial D_2} \frac{\partial W'(v, h)}{\partial D} + C \frac{\partial^2 W'(v, h)}{\partial D_2 \partial D}$

Cuadro A.5: Kernel $W(v, h)$: Derivadas. Cambiar variable genérica D por: $x_i, x_j; y_i, y_j; z_i, z_j; h_i, h_j$.

$\frac{\partial C}{\partial X} = 0$	$\frac{\partial C}{\partial H} = -C \frac{d}{h} \frac{\partial h}{\partial H}$
$\frac{\partial^2 C}{\partial H_2 \partial H} = -d \frac{\partial \left[\frac{C}{h} \right]}{\partial H_2} \frac{\partial h}{\partial H} = C \frac{d(d+1)}{h^2} \frac{\partial h}{\partial H_2} \frac{\partial h}{\partial H}$	$\frac{\partial \left[\frac{C}{h} \right]}{\partial H_2} = -C \frac{(d+1)}{h^2} \frac{\partial h}{\partial H_2}$

Cuadro A.6: Kernel Cubic Spline: Derivadas de C . Cambiar variable genérica X por: $x_i, x_j; y_i, y_j; z_i, z_j$. Cambiar variables genéricas H y H_2 por: h_i, h_j .

	$0 < v < 1$	$2 \geq v < 2$
$\mathbf{W}'(v, h) =$	$1 - \frac{3}{2}v^2 + \frac{3}{4}v^3$	$\frac{1}{4}(2-v)^3$
$\frac{\partial [\mathbf{W}'(v, h)]}{\partial D} =$	$\left[\frac{9}{4}v^2 - 3v\right] \frac{\partial v}{\partial D}$	$\left[-\frac{3}{4}(2-v)^2\right] \frac{\partial v}{\partial D}$
$\frac{\partial^2 [\mathbf{W}'(v, h)]}{\partial D_2 \partial D} =$	$\left[\frac{9}{2}v - 3\right] \frac{\partial v}{\partial D_2} \frac{\partial v}{\partial D} + \frac{\partial [\mathbf{W}'(v, h)]}{\partial D} \frac{\partial^2 v}{\partial D_2 \partial D}$	$\left[\frac{3}{2}(2-v)\right] \frac{\partial v}{\partial D_2} \frac{\partial v}{\partial D} + \frac{\partial [\mathbf{W}'(v, h)]}{\partial D} \frac{\partial^2 v}{\partial D_2 \partial D}$

Cuadro A.7: Kernel $\mathbf{W}'(v, h)$: Derivadas. Cambiar variable genérica D por: x_i, x_j ; y_i, y_j ; z_i, z_j ; h_i, h_j .

A.2. Kernel Sinc: $W^S n(v, h, n)$

$$W_n^S(v, h, n) = \frac{B_n}{h^d} S_n\left(\frac{\pi}{2}v\right) \quad (0 \leq v \leq 2) \quad (\text{A.2})$$

▪ n es el índice del kernel y B_n la constante de normalización.

▪ $S_n\left(\frac{\pi}{2}v\right) = \text{sinc}^n\left(\frac{\pi}{2}v\right)$

▪ $\text{sinc}\left(\frac{\pi}{2}v\right) = \left[\frac{\sin\left(\frac{\pi}{2}v\right)}{\left(\frac{\pi}{2}v\right)}\right]$

$$B_n = \begin{cases} b_0 + b_1 n^{1/2} + b_2 n + b_3 n^{-1/2}, & 1D \\ b_0 + b_1 n + b_2 n^{-1} + b_3 n^{-2}, & 2D \\ b_0 + b_1 n^{1/2} + b_2 n + b_3 n^{3/2}, & 3D \end{cases} \quad (\text{A.3})$$

Dimensiones	b_0	b_1	b_2	b_3
1D	$-1.5404568 \cdot 10^{-2}$	$3.6632876 \cdot 10^{-1}$	$-4.6519576 \cdot 10^{-4}$	$7.3658324 \cdot 10^{-2}$
2D	$5.2245027 \cdot 10^{-2}$	$1.3090245 \cdot 10^{-1}$	$1.9358485 \cdot 10^{-2}$	$-6.1642906 \cdot 10^{-3}$
3D	$2.7012593 \cdot 10^{-2}$	$2.0410827 \cdot 10^{-2}$	$3.7451957 \cdot 10^{-3}$	$4.7013839 \cdot 10^{-2}$

Cuadro A.8: Kernel Sinc: Constante de normalización B_n .

$\frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial D} =$	$\frac{\partial v}{\partial D} \left[\cos \left(\frac{\pi}{2} v \right) - \text{sinc} \left(\frac{\pi}{2} v \right) \right]$
$\frac{\partial^2 \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial D_2 \partial D} =$	$\left(\frac{\frac{\partial v}{\partial D_2} v - \frac{\partial v}{\partial D_2} \frac{\partial v}{\partial D}}{v^2} \right) \left[\cos \left(\frac{\pi}{2} v \right) - \text{sinc} \left(\frac{\pi}{2} v \right) \right] - \frac{\partial v}{\partial D} \left[\sin \left(\frac{\pi}{2} v \right) \frac{\pi}{2} \frac{\partial v}{\partial D_2} + \frac{\partial \text{sinc} \left(\frac{\pi}{2} v \right)}{\partial D_2} \right]$

Cuadro A.9: Kernel Sinc: Derivadas de *sinc*. Cambiar variables genéricas D , D_2 por: x_i, x_j ; y_i, y_j ; z_i, z_j ; h_i, h_j

$\frac{\partial [\mathbf{W}_n^S(v, h, n)]}{\partial X} =$	$n \left(\frac{\mathbf{W}_n^S(v, h_i, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial X} \right)$
$\frac{\partial^2 [\mathbf{W}_n^S(v, h, n)]}{\partial D \partial X} =$	$n \left(\frac{\frac{\partial \left[\frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \right]}{\partial D} \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial X} + \frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \frac{\partial^2 \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial D \partial X} \right)$
$\frac{\partial [\mathbf{W}_n^S(v, h, n)]}{\partial H} =$	$n \left(\frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial H} \right) - \frac{d}{h} \mathbf{W}_n^S(v, h, n) \frac{\partial h}{\partial H}$
$\frac{\partial^2 [\mathbf{W}_n^S(v, h, n)]}{\partial X_2 \partial H} =$	$n \left(\frac{\frac{\partial \left[\frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \right]}{\partial X_2} \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial H} + \frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \frac{\partial^2 \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial X_2 \partial H} \right) - \frac{d}{h} \left[\frac{\partial \mathbf{W}_n^S(v, h, n)}{\partial X_2} \right] \frac{\partial h}{\partial H}$
$\frac{\partial^2 [\mathbf{W}_n^S(v, h, n)]}{\partial H_2 \partial H} =$	$n \left(\frac{\frac{\partial \left[\frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \right]}{\partial H_2} \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial H} + \frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \frac{\partial^2 \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial H_2 \partial H} \right) - \frac{d}{h} \left[\frac{\partial \mathbf{W}_n^S(v, h, n)}{\partial H_2} - \frac{\mathbf{W}_n^S(v, h, n)}{h} \frac{\partial h}{\partial H_2} \right] \frac{\partial h}{\partial H}$
$\frac{\partial \left[\frac{\mathbf{W}_n^S(v, h, n)}{\text{sinc} \left(\frac{\pi}{2} v \right)} \right]}{\partial D} =$	$\frac{\frac{\partial [\mathbf{W}_n^S(v, h, n)]}{\partial D} \text{sinc} \left(\frac{\pi}{2} v \right) - \frac{\partial \left[\text{sinc} \left(\frac{\pi}{2} v \right) \right]}{\partial D} \mathbf{W}_n^S(v, h, n)}{\text{sinc}^2 \left(\frac{\pi}{2} v \right)}$

Cuadro A.10: Kernel Sinc: Derivadas de S_n . Cambiar variables genéricas X y X_2 por: x_i, x_j ; y_i, y_j ; z_i, z_j . Cambiar variables genéricas H y H_2 por: h_i, h_j . Cambiar variable genérica D por: x_i, x_j ; y_i, y_j ; z_i, z_j ; h_i, h_j .

Apéndice B

Derivadas parciales de las ecuaciones físicas

Como se especifica en la sección 2.3.4, en las ecuaciones que se utilizan en el método *Newton-Raphson* se busca que se aproximen lo máximo a cero. En los siguientes cuadros se puede observar las derivadas de las ecuaciones utilizadas en el *Newton-Raphson*, según el siguiente desglose:

- Ecuación del momento: $G_{a_x}, G_{a_y}, G_{a_z}$ (B.1)
- Ecuación de la energía o temperatura: G_E (B.2)
- Ecuación de la distancia de interpolacion: G_h (B.4)
- Ecuación del movimiento: $G_{v_z}, G_{v_y}, G_{v_x}$ (B.5)
- Ecuación de la densidad: G_ρ (B.6)

Se van a tomar una serie de aproximaciones que no tienen un impacto significativo en la precisión y que simplifican la derivación. Estas aproximaciones sólo se utilizan en el *Newton-Raphson*, es decir, que afectan a la rapidez de convergencia del cálculo, pero no a la exactitud del resultado. Además estas aproximaciones permiten mejorar enormemente la distribución de la matriz a invertir, reduciendo la dispersión de la misma, una ventaja que compensa de sobras la posibilidad de necesitar mas refinamientos.

Tomaremos el siguiente criterio práctico:

$$\frac{\partial P_i}{\partial T_i} = cte \quad (B.1)$$

$$C_v = \frac{\partial U_i}{\partial T_i} = cte \quad (B.2)$$

Podemos encontrar las ecuaciones G_i^n completas en el cuadro 2.8 de la sección 2.2.9. Si observamos la estructura de una ecuación cualquiera, para el esquema centrado, podemos ver que habitualmente contiene:

- Un término positivo que hace referencia a una variación temporal.
- Un término negativo que hace referencia al valor que se ha calculado anteriormente ($n - 1$) de la variable independiente acompañado del parámetro $(1 - \theta)$.
- Un término negativo que hace referencia al valor que se está calculando actualmente (n) de la variable independiente acompañado del parámetro (θ) .

Esto no es así en la ecuación de la densidad (ecuación 2.136 y derivadas cuadro B.6), ya que depende únicamente del valor actual de la variable independiente (n) y no del valor en el tiempo anterior ($n - 1$). En este caso, por estandarización que luego será útil a la hora de implementar el código, se aproxima a la estructura descrita del siguiente modo:

- El término positivo que hace referencia a una variación temporal se sustituye por el valor de la variable, en este caso ρ_i .
- El término negativo que hace referencia al valor que se ha calculado anteriormente acompañado del parámetro $(1 - \theta)$ se hace 0.
- El término negativo que hace referencia al valor que se está calculando actualmente de la variable independiente acompañado de (θ) , se simula colocando el factor θ multiplicando y dividiendo de tal modo que se puede colocar en el mismo formato que el resto de ecuaciones sin modificar el valor del término. Lo mismo se hace con el signo, multiplicando dos veces por el número -1 y obteniendo por tanto el factor $-\theta$ que se necesita.

Por último, se observa que en los cuadros aparecen los valores y las derivadas de componentes de física adicional (gravedad, viscosidad y conducción térmica), podemos encontrar la definición de esos términos en el apéndice C.

En adelante, y por claridad en las ecuaciones, se toman variables genéricas (como X , H , E , *etc.*) que pueden ser sustituidas por variables concretas (como x_i , z_j , h_i , U_j *etc.*). Para obtener todas las ecuaciones reales hay que realizar todas las combinaciones de variables. En la leyenda de cada ecuación se especifican las opciones disponibles.

$G_{a_k} =$	$\frac{v_{k_i} - v_{k_i}^{(n-1)}}{\Delta t} - (1 - \theta) a_k^{(n-1)} - \theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial x_i} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial x_i} \right) + a_k^{visco} + a_k^{grav} + a_k^{damp} \right]$
$\frac{\partial G_{a_k}}{\partial X} \cdot \delta X =$	$-\theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \left\{ \frac{\partial^2 W_{ij}(h_i)}{\partial X \partial x_i} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial X} \frac{\partial W_{ij}(h_i)}{\partial x_i} \right\} + \frac{P_j}{\Omega_j \rho_j^2} \left\{ \frac{\partial^2 W_{ij}(h_j)}{\partial X \partial x_i} - \frac{1}{\Omega_j} \frac{\partial \Omega_j}{\partial X} \frac{\partial W_{ij}(h_j)}{\partial x_i} \right\} \right) + \frac{\partial a_k^{visco}}{\partial X} + \frac{\partial a_k^{grav}}{\partial X} + \frac{\partial a_k^{damp}}{\partial X} \right] \cdot \delta X$
$\frac{\partial G_{a_k}}{\partial E} \cdot \delta E =$	$-\theta \left[-\sum_{j=1}^N m_j \left(\frac{1}{\Omega_i \rho_i^2} \left(\frac{\partial P_i}{\partial E} \right) \frac{\partial W_{ij}(h_i)}{\partial x_i} + \frac{1}{\Omega_j \rho_j^2} \left(\frac{\partial P_j}{\partial E} \right) \frac{\partial W_{ij}(h_j)}{\partial x_i} \right) + \frac{\partial a_k^{visco}}{\partial E} + \frac{\partial a_k^{grav}}{\partial E} + \frac{\partial a_k^{damp}}{\partial E} \right] \cdot \delta E$
$\frac{\partial G_{a_k}}{\partial h} \cdot \delta h =$	$-\theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \left\{ \left(\frac{1}{P_i} \frac{\partial P_i}{\partial h} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial h} \right) \frac{\partial W_{ij}(h_i)}{\partial x_i} + \frac{\partial^2 W_{ij}(h_i)}{\partial h \partial x_i} \right\} + \frac{P_j}{\Omega_j \rho_j^2} \left\{ \left(\frac{1}{P_j} \frac{\partial P_j}{\partial h} - \frac{1}{\Omega_j} \frac{\partial \Omega_j}{\partial h} \right) \frac{\partial W_{ij}(h_j)}{\partial x_i} + \frac{\partial^2 W_{ij}(h_j)}{\partial h \partial x_i} \right\} \right) + \frac{\partial a_k^{visco}}{\partial h} + \frac{\partial a_k^{grav}}{\partial h} + \frac{\partial a_k^{damp}}{\partial h} \right] \cdot \delta h$
$\frac{\partial G_{a_k}}{\partial v_X} \cdot \delta v_X =$	$-\theta \left[-\frac{1}{\theta} \frac{1}{\Delta t} \frac{\partial v_{k_i}}{\partial v_X} + \frac{\partial a_k^{visco}}{\partial v_X} + \frac{\partial a_k^{grav}}{\partial v_X} + \frac{\partial a_k^{damp}}{\partial v_X} \right] \cdot \delta v_X$
$\frac{\partial G_{a_k}}{\partial \rho} \cdot \delta \rho =$	$-\theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \left\{ \frac{1}{P_i} \frac{\partial P_i}{\partial \rho} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial \rho} - \frac{2}{\rho_i} \frac{\partial \rho_i}{\partial \rho} \right\} \frac{\partial W_{ij}(h_i)}{\partial x_i} + \frac{P_j}{\Omega_j \rho_j^2} \left\{ \frac{1}{P_j} \frac{\partial P_j}{\partial \rho} - \frac{1}{\Omega_j} \frac{\partial \Omega_j}{\partial \rho} - \frac{2}{\rho_j} \frac{\partial \rho_j}{\partial \rho} \right\} \frac{\partial W_{ij}(h_j)}{\partial x_i} \right) + \frac{\partial a_k^{visco}}{\partial \rho} + \frac{\partial a_k^{grav}}{\partial \rho} + \frac{\partial a_k^{damp}}{\partial \rho} \right] \cdot \delta \rho$

Cuadro B.1: Contiene las derivas de la ecuación del momento G_{a_k} , donde k son las ecuaciones según las direcciones (x, y, z) . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

$G_{\dot{E}} =$	$\frac{E_i - E_i^{(n-1)}}{\Delta t} - (1 - \theta) \dot{E}^{(n-1)} - \theta \left[\frac{e_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i W_{ij}(h_i) + \dot{E}_{visco} + \dot{E}_{cond} + \dot{E}_{nuc} \right]$
$\frac{\partial G_{\dot{E}}}{\partial X} \cdot \delta X =$	$-\theta \left[\frac{e_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \left\{ \frac{\partial \nabla_i W_{ij}(h_i)}{\partial X} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial X} \nabla_i W_{ij}(h_i) \right\} + \frac{\partial \dot{E}_{visco}}{\partial X} + \frac{\partial \dot{E}_{cond}}{\partial X} + \frac{\partial \dot{E}_{nuc}}{\partial X} \right] \cdot \delta X$
$\frac{\partial G_{\dot{E}}}{\partial E} \cdot \delta E =$	$-\theta \left[-\frac{1}{\theta} \frac{1}{\Delta t} \frac{\partial E_i}{\partial E} + \frac{1}{\Omega_i \rho_i^2} \frac{\partial e_i}{\partial E} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i W_{ij}(h_i) + \frac{\partial \dot{E}_{visco}}{\partial E} + \frac{\partial \dot{E}_{cond}}{\partial E} + \frac{\partial \dot{E}_{nuc}}{\partial E} \right] \cdot \delta E$
$\frac{\partial G_{\dot{E}}}{\partial h} \cdot \delta h =$	$-\theta \left[\frac{e_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \left\{ \frac{\partial \nabla_i W_{ij}(h_i)}{\partial h} + \left(\frac{1}{e_i} \frac{\partial e_i}{\partial h} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial h} \right) \nabla_i W_{ij}(h_i) \right\} + \frac{\partial \dot{E}_{visco}}{\partial h} + \frac{\partial \dot{E}_{cond}}{\partial h} + \frac{\partial \dot{E}_{nuc}}{\partial h} \right] \cdot \delta h$
$\frac{\partial G_{\dot{E}}}{\partial v_X} \cdot \delta v_X =$	$-\theta \left[\frac{e_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j \frac{\partial (v_{X_i} - v_{X_j})}{\partial v_X} \frac{\partial W_{ij}(h_i)}{\partial X} + \frac{\partial \dot{E}_{visco}}{\partial v_X} + \frac{\partial \dot{E}_{cond}}{\partial v_X} + \frac{\partial \dot{E}_{nuc}}{\partial v_X} \right] \cdot \delta v_X$
$\frac{\partial G_{\dot{E}}}{\partial \rho} \cdot \delta \rho =$	$-\theta \left[\frac{e_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \left(\frac{1}{e_i} \frac{\partial e_i}{\partial \rho} - \frac{1}{\Omega_i} \frac{\partial \Omega_i}{\partial \rho} - \frac{2}{\rho_i} \frac{\partial \rho_i}{\partial \rho} \right) \nabla_i W_{ij}(h_i) + \frac{\partial \dot{E}_{visco}}{\partial \rho} + \frac{\partial \dot{E}_{cond}}{\partial \rho} + \frac{\partial \dot{E}_{nuc}}{\partial \rho} \right] \cdot \delta \rho$

Cuadro B.2: Contiene las derivas de la ecuación $G_{\dot{E}}$. Donde, si $(G_{\dot{E}} = G_{\dot{U}}) \Rightarrow (e_i = P_i)$ y si $(G_{\dot{E}} = G_{\dot{T}}) \Rightarrow \left(e_i = \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \right)$ Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

	$G_{\dot{U}}$	$G_{\dot{T}}$
$e_i =$	P_i	$\frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v}$
$\frac{\partial e_i}{\partial T} =$	(No aplica)	$\frac{\partial P_i}{\partial T_i} \frac{1}{C_v} \frac{\partial T_i}{\partial T}$
$\frac{\partial e_i}{\partial U} =$	$\frac{\partial P_i}{\partial U}$	(No aplica)
$\frac{\partial e_i}{\partial h} =$	$\frac{\partial P_i}{\partial h}$	(Derivada aproximada a 0)
$\frac{\partial e_i}{\partial \rho} =$	$\frac{\partial P_i}{\partial \rho}$	(Derivada aproximada a 0)

Cuadro B.3: Contiene las derivas de la variable e_i . Cambiar variables genéricas $T = T_i, T_j$; $U = U_i, U_j$; $h = h_i, h_j$; $\rho = \rho_i, \rho_j$.

$G_h =$	$K h_i^{-d} - \theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j W_{ij}(h_i) \right]$
$\frac{\partial G_h}{\partial X} \cdot \delta X =$	$-\theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial X} \right] \cdot \delta X$
$\frac{\partial G_h}{\partial E} \cdot \delta E =$	0
$\frac{\partial G_h}{\partial h} \cdot \delta h =$	$-\theta \left[\frac{1}{\theta} \left(K h_i^{-d} \frac{d}{h_i} \frac{\partial h_i}{\partial h} + \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h} \right) \right] \cdot \delta h$
$\frac{\partial G_h}{\partial v_X} \cdot \delta v_X =$	0
$\frac{\partial G_h}{\partial \rho} \cdot \delta \rho =$	0

Cuadro B.4: Contiene las derivas de la ecuación G_h . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

$G_{v_k} =$	$\frac{k_i - k_i^{(n-1)}}{\Delta t} - (1 - \theta) v_k^{(n-1)} - \theta \left[\frac{v_{ki} + v_{ki}^{(n-1)}}{2} + \varepsilon \sum_{j=1}^N m_j \frac{(v_{kj} - v_{ki})}{\bar{\rho}_{ij}} W_{ij}(h_i) \right]$
$\frac{\partial G_{v_k}}{\partial X} \cdot \delta X =$	$-\theta \left[-\frac{1}{\theta} \frac{1}{\Delta t} \frac{\partial k_i}{\partial X} + \varepsilon \sum_{j=1}^N m_j \frac{(v_{kj} - v_{ki})}{\bar{\rho}_{ij}} \frac{\partial W_{ij}(h_i)}{\partial X} \right] \cdot \delta X$
$\frac{\partial G_{v_k}}{\partial E} \cdot \delta E =$	0
$\frac{\partial G_{v_k}}{\partial h} \cdot \delta h =$	$-\theta \left[\varepsilon \sum_{j=1}^N m_j \frac{(v_{kj} - v_{ki})}{\bar{\rho}_{ij}} \frac{\partial W_{ij}(h_i)}{\partial h} \right] \cdot \delta h$
$\frac{\partial G_{v_k}}{\partial v_X} \cdot \delta v_X =$	$-\theta \left[\frac{1}{2} \frac{\partial v_{ki}}{\partial v_X} + \varepsilon \sum_{j=1}^N m_j \frac{1}{\bar{\rho}_{ij}} \frac{\partial (v_{kj} - v_{ki})}{\partial v_X} W_{ij}(h_i) \right] \cdot \delta v_X$
$\frac{\partial G_{v_k}}{\partial \rho} \cdot \delta \rho =$	$-\theta \left[-\varepsilon \sum_{j=1}^N m_j \frac{(v_{kj} - v_{ki})}{2 (\bar{\rho}_{ij})^2} \frac{\partial (\rho_i + \rho_j)}{\partial \rho} W_{ij}(h_i) \right] \cdot \delta \rho$

Cuadro B.5: Contiene las derivas de la ecuación G_{v_k} , donde k son las ecuaciones según las direcciones (x, y, z) . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j ; $h = h_i, h_j$; $\rho = \rho_i, \rho_j$

$G_{[\rho//\dot{\rho}]} =$	$G_{\rho} = \rho_i - \theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j W_{ij}(h_i) \right]$	$G_{\dot{\rho}_i} = \frac{\rho_i - \rho_i^{(n-1)}}{\Delta t} - (1 - \theta) \dot{\rho}^{(n-1)} - \theta \left[\sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right]$
$\frac{\partial G_{[\rho//\dot{\rho}]}}{\partial X} \cdot \delta X =$	$-\theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial X} \right] \cdot \delta X$	$-\theta \left[\sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \frac{\partial \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial X} \right] \cdot \delta X$
$\frac{\partial G_{[\rho//\dot{\rho}]}}{\partial E} \cdot \delta E =$	0	0
$\frac{\partial G_{[\rho//\dot{\rho}]}}{\partial h} \cdot \delta h =$	$-\theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j \frac{\partial W_{ij}(h_i)}{\partial h} \right] \cdot \delta h$	$-\theta \left[\sum_{j=1}^N m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \frac{\partial \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial h} \right] \cdot \delta h$
$\frac{\partial G_{[\rho//\dot{\rho}]}}{\partial \rho} \cdot \delta \rho =$	$-\theta \left[-\frac{1}{\theta} \frac{\partial \rho_i}{\partial \rho} \right] \cdot \delta \rho$	$-\theta \left[-\frac{1}{\theta} \frac{1}{\Delta t} \frac{\partial \rho_i}{\partial \rho} \right] \cdot \delta \rho$
$\frac{\partial G_{[\rho//\dot{\rho}]}}{\partial v_X} \cdot \delta v_X =$	0	$-\theta \left[\sum_{j=1}^N m_j \frac{\partial (v_i^X - v_j^X)}{\partial v_X} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial X_i} \right] \cdot \delta v_X$

Cuadro B.6: Contiene las derivas de la ecuación G_{ρ} . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

Apéndice C

Física adicional

Como se especifica en la sección 2.2, es necesario agregar física adicional a las ecuaciones para poder reproducir modelos que se aproximen a la realidad. En este apartado veremos las derivadas necesarias de las piezas de física adicional que se han implementado en el código ISFAA: Viscosidad (C.1), Gravedad (C.2), Conducción térmica (C.3) y Damping - Término de amortiguamiento artificial (C.15), solo utilizado para relajar modelos en equilibrio. .

En adelante, y por claridad en las ecuaciones, se toman variables genéricas (como X , E , *etc.*) que pueden ser sustituidas por variables concretas (como x_i , z_j , U_i , T_j *etc.*). Para obtener todas las ecuaciones reales hay que realizar todas las combinaciones de variables. En la leyenda de cada ecuación se especifican las opciones disponibles.

C.1. Viscosidad Artificial (Π_{ij})

Como se especifica en la sección 2.2.5, la viscosidad artificial en SPH sirve para agregar de forma eficiente disipación en las zonas de choque y evitar al máximo la introducción de ruido numérico en el modelo.

La implementación que se utiliza en el código es la más estándar (*Monaghan, 1992* [58], ecuación C.1). Como podemos observar su aplicación se encuentra determinada por la relación entre velocidad y posición relativa entre las partículas i, j .

Por último se aplicará la definición de viscosidad artificial (ecuación C.1) y sus derivadas (cuadro C.3) a la ecuación del momento (cuadro C.4) y la ecuación de la energía/temperatura (cuadro C.5).

$$\Pi_{ij} = \begin{cases} \frac{-\alpha \bar{C}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}}, & (\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0) \\ 0, & (\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} \geq 0) \end{cases} \quad (\text{C.1})$$

$$\alpha = 1$$

$$\beta = 2$$

$$C_s = \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_S}$$

$$\bar{C}_{sij} = \frac{C_{si} + C_{sj}}{2}$$

$$\bar{\rho}_{ij} = \frac{\rho_i + \rho_j}{2}$$

$$\bar{h}_{ij} = \frac{h_i + h_j}{2}$$

$$r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$$

$$\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} = (v_{xi} - v_{xj})(x_i - x_j) + (v_{yi} - v_{yj})(y_i - y_j) + (v_{zi} - v_{zj})(z_i - z_j)$$

Cuadro C.1: Viscosidad artificial: Formulación estándar. Donde \bar{C}_{ij} es la velocidad del sonido media, $\bar{\rho}_{ij}$ es la densidad media, α es el parámetro de escalado del término lineal, β es el parámetro de escalado del término cuadrático y η_μ previene singularidades. Se toma la aproximación $C_s = cte$ para simplificar la derivación

$\mu_{ij} = \bar{h}_{ij} \cdot \frac{(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})}{r_{ij}^2 + \eta_\mu^2}$	$\eta_\mu = 0.1 \cdot \bar{h}_{ij}$
$\frac{\partial \mu_{ij}}{\partial X} =$	$\mu_{ij} \left(\frac{v_{X_i} - v_{X_j}}{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}} - \frac{2 (X_i - X_j)}{r_{ij}^2 + \eta_\mu^2} \right) \frac{\partial [X_i - X_j]}{\partial X}$
$\frac{\partial \mu_{ij}}{\partial v_X} =$	$\mu_{ij} \left(\frac{X_i - X_j}{\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}} \right) \frac{\partial [v_{i_X} - v_{j_X}]}{\partial v_X}$
$\frac{\partial \mu_{ij}}{\partial h} =$	$\mu_{ij} \left(\frac{1}{2 \bar{h}_{ij}} - \frac{0.1 \eta_\mu}{r_{ij}^2 + \eta_\mu^2} \right)$

Cuadro C.2: Viscosidad artificial: Derivadas μ_{ij} . Cambiar variable genérica X por: x_i, x_j ; y_i, y_j ; z_i, z_j .

$\Pi_{ij} =$	$\frac{-\alpha \bar{C}_{ij} \mu_{ij} + \beta \mu_{ij}^2}{\bar{\rho}_{ij}}$
$\frac{\partial \Pi_{ij}}{\partial X} =$	$\left[\frac{-\alpha \bar{C}_{ij} + 2 \beta \mu_{ij}}{\bar{\rho}_{ij}} \right] \frac{\partial \mu_{ij}}{\partial X}$
$\frac{\partial \Pi_{ij}}{\partial E} =$	$-\frac{\alpha \mu_{ij}}{2 \bar{\rho}_{ij}} \frac{\partial [C_{s_i} + C_{s_j}]}{\partial E}$
$\frac{\partial \Pi_{ij}}{\partial h} =$	$\left[\frac{-\alpha \bar{C}_{ij} + 2 \beta \mu_{ij}}{\bar{\rho}_{ij}} \right] \frac{\partial \mu_{ij}}{\partial h}$
$\frac{\partial \Pi_{ij}}{\partial v_X} =$	$\left[\frac{-\alpha \bar{C}_{ij} + 2 \beta \mu_{ij}}{\bar{\rho}_{ij}} \right] \frac{\partial \mu_{ij}}{\partial v_X}$
$\frac{\partial \Pi_{ij}}{\partial \rho} =$	$-\frac{1}{2} \frac{\Pi_{ij}}{\bar{\rho}_{ij}}$

Cuadro C.3: Viscosidad artificial: Π_{ij} . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

$a_{visco}^k =$	$-\sum_{j=1}^N m_j \Pi_{ij} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k}$
$\frac{\partial a_{visco}^k}{\partial X} =$	$-\sum_{j=1}^N m_j \left(\frac{\partial \Pi_{ij}}{\partial X} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k} + \Pi_{ij} \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial X \partial k} \right)$
$\frac{\partial a_{visco}^k}{\partial E} =$	$-\sum_{j=1}^N m_j \left(\frac{\partial \Pi_{ij}}{\partial E} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k} \right)$
$\frac{\partial a_{visco}^k}{\partial h} =$	$-\sum_{j=1}^N m_j \left(\frac{\partial \Pi_{ij}}{\partial h} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k} + \Pi_{ij} \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial h \partial k} \right)$
$\frac{\partial a_{visco}^k}{\partial v_X} =$	$-\sum_{j=1}^N m_j \left(\frac{\partial \Pi_{ij}}{\partial v_X} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k} \right)$
$\frac{\partial a_{visco}^k}{\partial \rho} =$	$-\sum_{j=1}^N m_j \left(\frac{\partial \Pi_{ij}}{\partial \rho} \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial k} \right)$

Cuadro C.4: Viscosidad artificial: Ecuación del momento, donde k son las ecuaciones según las direcciones (x, y, z) . Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

$\dot{T}_{visco} = \frac{1}{C_v} \dot{U}_{visco} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j (v_i - v_j) \Pi_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$
$\frac{\partial \dot{T}_{visco}}{\partial X} = \frac{1}{C_v} \frac{\partial \dot{U}_{visco}}{\partial X} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j (v_i - v_j) \left(\frac{\partial \Pi_{ij}}{\partial X} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) + \Pi_{ij} \frac{\partial \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial X} \right)$
$\frac{\partial \dot{T}_{visco}}{\partial E} = \frac{1}{C_v} \frac{\partial \dot{U}_{visco}}{\partial E} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j (v_i - v_j) \frac{\partial \Pi_{ij}}{\partial E} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$
$\frac{\partial \dot{T}_{visco}}{\partial h} = \frac{1}{C_v} \frac{\partial \dot{U}_{visco}}{\partial h} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j (v_i - v_j) \left(\frac{\partial \Pi_{ij}}{\partial h} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) + \Pi_{ij} \frac{\partial \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial h} \right)$
$\frac{\partial \dot{T}_{visco}}{\partial v_X} = \frac{1}{C_v} \frac{\partial \dot{U}_{visco}}{\partial v_X} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j \left((v_i - v_j) \frac{\partial \Pi_{ij}}{\partial v_X} + \Pi_{ij} \frac{\partial (v_i - v_j)}{\partial v_X} \right) \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$
$\frac{\partial \dot{T}_{visco}}{\partial \rho} = \frac{1}{C_v} \frac{\partial \dot{U}_{visco}}{\partial \rho} =$	$\frac{1}{C_v} \frac{1}{2} \sum_{j=1}^N m_j (v_i - v_j) \frac{\partial \Pi_{ij}}{\partial \rho} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$

Cuadro C.5: Viscosidad artificial: Ecuación de la energía/temperatura, donde k son las ecuaciones según las direcciones (x, y, z) . $\dot{T}_{visco} = \dot{U}_{visco}/C_v$. Si $C_v = cte$ tenemos la relación $Deriv \dot{T}_{visco} = 1/C_v \cdot \dot{T}_{visco}$. Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j .

C.2. Gravedad (g_i)

Como se especifica en la sección 2.2.4, en este apartado se muestran las cuatro implementaciones de la gravedad que pueden ser elegidas en ISFAA, así como sus definiciones y derivadas.

Estas implementaciones las podemos identificar como: Gravedad directa, gravedad según la ley de Gauss y gravedad según el potencial gravitatorio. En el código de forma general se escoge la gravedad aproximada por el potencial gravitatorio calculado mediante la aproximación cuadrupolar, pero sin perder de vista las limitaciones de la implementación, que se comentarán más adelante, y comprobando la desviación del resultado con la gravedad directa.

En astrofísica, en general, se usa el sistema *cgs* para las unidades físicas y por tanto la constante de gravitación universal usada en estas ecuaciones de la gravedad tiene un valor de:

$$G = 6,6726 \cdot 10^{-8} \text{ cm}^3 \text{ g}^{-1} \text{ s}^{-2} \quad (\text{C.2})$$

Ahora pasemos ahora a ver las implementaciones en detalle.

Gravedad según un oscilador armónico

$\mathbf{a}_{\text{grav}} =$	$-\kappa \cdot \left \mathbf{r}^{CM} \right ^n \cdot \widehat{\mathbf{r}^{CM}}$
$a_{\text{grav}}^k =$	$-\kappa \cdot \left \mathbf{r}^{CM} \right ^n \cdot \frac{(k - k^{CM})}{r^{CM}}$
$\frac{\partial a_{\text{grav}}^k}{\partial X} =$	$-\kappa \left[\left(r^{CM} \right)^{n-1} + (n-1) \left(X - X^{CM} \right) \left(k - k^{CM} \right) \left(r^{CM} \right)^{n-3} \right] \frac{\partial k}{\partial x}$

Cuadro C.6: Gravedad según un oscilador armónico: Ecuación del momento, donde k son las variables según las direcciones (x_i, y_i, z_i) , k^{CM} son las direcciones del centro de masas (x^{CM}, y^{CM}, z^{CM}) , κ es una constante y n es el índice del oscilador. Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

Cuando $\kappa = 1$ y $n = 1$ tenemos que el oscilador armónico se comporta como una gravedad que depende de la distancia al centro de masas. Esto es utilizado en el test evolución casi-estática de un sistema próximo al equilibrio (Toy Star).

Gravedad directa: Interacción directa entre partículas (método explícito)

La gravedad directa sirve para todo tipo de modelos y es exacta, pero computacionalmente cara, con un coste $\theta(n^2)$. Además solo es recomendable usarla en su forma explícita, es decir solo debe influir en las ecuaciones centradas en el término explícito que por convenio se ha asociado al término que va con $(1 - \theta)$. Si lo utilizamos de forma implícita, como no tiene en su fórmula el gradiente del kernel, aparecería en toda la matriz (en las derivadas de todas las ecuaciones respecto de las posiciones $\frac{\partial^j G_i}{\partial x}, \frac{\partial^j G_i}{\partial y}, \frac{\partial^j G_i}{\partial z}$ y respecto de la masa $\frac{\partial^j G_i}{\partial m}$), convirtiendo una matriz dispersa en una matriz general que al resolver tiene un coste computacional inaccesible.

$\mathbf{a}_{\text{grav}}^i = \frac{\mathbf{F}_{\mathbf{g}i}}{m_i} = -G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^2} \hat{\mathbf{r}}_{ij} = -G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^3} \mathbf{r}_{ij}$
$a_{\text{grav}}^{k_i} = -G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^3} (k_i - k_j)$

Cuadro C.7: Gravedad directa: Método explícito

Gravedad según ley de Gauss: Simetría radial (método implícito)

La gravedad según la ley de Gauss se utiliza para sistemas en los que se pueda suponer simetría radial. En sistemas reales es aproximada pero computacionalmente aceptable, con un coste $\theta(n \cdot \log n)$. Si se pierde la simetría deja de ser válida. Su cálculo necesita la ordenación de las partículas según el radio y calcula la gravedad suponiendo que las partículas están repartidas de forma homogénea en el modelo. Puede ser implementada de forma implícita ya que al derivar el radio al centro de masas r_i^{CM} y la masa encerrada en el radio $M(r_i^{CM})$ solo depende de las posiciones de i y al derivar respecto de las partículas j se vuelve 0. Por tanto no tenemos el problema anterior de la gravedad directa que provocaba la pérdida de dispersión en la matriz.

Las derivadas de la función $M(r_i^{CM})$, respecto de las posiciones, se calculan de forma tabulada mediante una aproximación cuadrática de tres partículas contiguas en relación a su distancia al centro de masas.

La aproximación cuadrática sigue la siguiente definición: $M(r_i^{CM}) = a(r_i^{CM})^2 + b(r_i^{CM}) + c$

La ecuación de gravedad según la ley de Gauss varía en función del número de dimensiones, mostrándose en los cuadros siguientes: 2D Cuadro C.8 y 3D Cuadro C.9.

$a_{grav}^i = 2 G \frac{M (r_i^{CM})}{r_i^{CM}} \frac{\mathbf{r}_i^{CM}}{r_i^{CM}} = 2 G \left[M (r_i^{CM}) \frac{\mathbf{r}_i^{CM}}{(r_i^{CM})^2} \right]$
$a_{grav}^{k_i} = 2 G \left[M (r_i^{CM}) \frac{k_i - k^{CM}}{(r_i^{CM})^2} \right]$
$\frac{\partial a_{grav}^{k_i}}{\partial X} = 2 G \left[\frac{\partial [M (r_i^{CM})]}{\partial X} \frac{k_i - k^{CM}}{(r_i^{CM})^2} + M (r_i^{CM}) \frac{\frac{\partial k_i}{\partial X} (r_i^{CM})^2 - 2 (X_i - X^{CM}) (k_i - k^{CM})}{(r_i^{CM})^4} \right]$

Cuadro C.8: Gravedad según la ley de Gauss: 2D (x,y), donde k son las variables según las direcciones (x_i, y_i) y k^{CM} son las direcciones del centro de masas (x^{CM}, y^{CM}) . Cambiar variables genéricas X por: $x_i, x_j; y_i, y_j$

$a_{grav}^i = G \frac{M (r_i^{CM})}{(r_i^{CM})^2} \frac{\mathbf{r}_i^{CM}}{r_i^{CM}} = G \left[M (r_i^{CM}) \frac{\mathbf{r}_i^{CM}}{(r_i^{CM})^3} \right]$
$a_{grav}^{k_i} = G \left[M (r_i^{CM}) \frac{k_i - k^{CM}}{(r_i^{CM})^3} \right]$
$\frac{\partial a_{grav}^{k_i}}{\partial X} = G \left[\frac{\partial [M (r_i^{CM})]}{\partial X} \frac{k_i - k^{CM}}{(r_i^{CM})^3} + M (r_i^{CM}) \frac{\frac{\partial k_i}{\partial X} (r_i^{CM})^3 - 3 (X_i - X^{CM}) (k_i - k^{CM})}{(r_i^{CM})^5} \right]$

Cuadro C.9: Gravedad según la ley de Gauss: 3D (x,y,z), , donde k son las variables según las direcciones (x_i, y_i, z_i) y k^{CM} son las direcciones del centro de masas (x^{CM}, y^{CM}, z^{CM}) . Cambiar variables genéricas X por: $x_i, x_j; y_i, y_j; z_i, z_j$

Gravedad según el potencial gravitatorio (método pseudo-implícito)

La gravedad según el potencial gravitatorio sirve para todo tipo de modelos.

Primeramente se calcula el potencial gravitatorio exacto mediante la ecuación 2.89 de forma explícita, ya que depende de las posiciones actuales y estas se toman como un número constante que se hace 0 cuando se deriva. El convenio $\phi = cte$ al ejecutar el *Newton-Raphson* también permite el uso de la expansión multipolar para calcular el potencial de forma aproximada según la ecuación 2.97.

Si el potencial gravitatorio se calcula de forma exacta, la gravedad tiene una buena aproximación pero el cálculo es computacionalmente caro, con un coste de $\theta(n^2)$ con lo que es preferible el uso de gravedad directa de forma explícita. Si el potencial gravitatorio se aproxima mediante el momento cuadrupolar el coste se reduce a $\theta(n \cdot \log n)$

Posteriormente se calcula la gravedad aproximada mediante la ecuación 2.95 de forma implícita, utilizando el potencial gravitatorio de los vecinos para calcular la gravedad de la partícula. Al tomar $\phi = cte$ y estar multiplicado por el kernel, se evita la pérdida de dispersión producida en la versión de gravedad directa, manteniendo los bloques de partículas no vecinos rellenos de 0s y obteniendo una contribución pseudo-implícita a la ecuación del momento.

De otra parte es necesario hacer notar que la ecuación 2.95 tiene una serie de defectos que es necesario valorar:

- Hay un defecto de simetría debido a la resta de potenciales y por tanto no conserva totalmente el momento lineal.
- Si se usa la aproximación cuadrupolar para el potencial gravitatorio, en la aproximación para partículas muy cercanas ($r_{ij} \leq 1 h$), introduce divergencias que se pueden minimizar colocando un parámetro de masa variable en el cálculo del potencial de los vecinos o como se ha indicado en la sección 2.2.4 calculando la gravedad de esas partículas mediante la interacción directa.

Pasemos a ver las ecuaciones necesarias para la implementación del método.

Potencial gravitatorio El potencial gravitatorio y sus derivadas se encuentran definidas según el cuadro C.10. Si utilizamos el suavizado de masas, se sustituye la masa que depende de j una masa que depende de i, j denominada m_{ij} (ecuación 2.107), para ello se multiplica la masa m_j por un factor η_m (ecuación 2.106).

$\phi_i = G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}}$	$k = i$	$k = j \wedge j \neq i$
$\frac{\partial \phi_i}{\partial x_k} =$	$-G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^3} (x_i - x_j)$	$G \frac{m_{ij}}{r_{ij}^3} (x_i - x_j)$
$\frac{\partial \phi_i}{\partial y_k} =$	$-G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^3} (y_i - y_j)$	$G \frac{m_{ij}}{r_{ij}^3} (y_i - y_j)$
$\frac{\partial \phi_i}{\partial z_k} =$	$-G \sum_{j=1}^n \frac{m_{ij}}{r_{ij}^3} (z_i - z_j)$	$G \frac{m_{ij}}{r_{ij}^3} (z_i - z_j)$

Cuadro C.10: Gravedad según el potencial gravitatorio: Cálculo del potencial gravitatorio.

Gravedad semi-implícita y derivadas: $\mathbf{a}_{\text{grav}} = \mathbf{g}_i = -\nabla_i \phi_i$ La gravedad obtenida con el potencial (ecuación 2.95) según las direcciones se encuentra definida en el cuadro C.11.

$$a_{grav}^X = \frac{1}{\rho_i} \sum_{j=1}^N m_j (\phi_i - \phi_j) \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial X}$$

$\frac{\partial a_{grav}^X}{\partial x_k} =$	$\frac{1}{\rho_i} \sum_{j=1}^N m_j \left[\left(\frac{\partial \phi_i}{\partial x_i} - \frac{\partial \phi_j}{\partial x_i} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial X} + (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial x_i \partial X} \right]$	$\frac{1}{\rho_i} m_j \left[\left(\frac{\partial \phi_i}{\partial x_j} - \frac{\partial \phi_j}{\partial x_j} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial x_i} - (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial x_i \partial x_i} \right]$
$\frac{\partial a_{grav}^X}{\partial y_k} =$	$\frac{1}{\rho_i} \sum_{j=1}^N m_j \left[\left(\frac{\partial \phi_i}{\partial y_i} - \frac{\partial \phi_j}{\partial y_i} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial X} + (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial y_i \partial X} \right]$	$\frac{1}{\rho_i} m_j \left[\left(\frac{\partial \phi_i}{\partial y_j} - \frac{\partial \phi_j}{\partial y_j} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial x_i} - (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial x_i \partial y_i} \right]$
$\frac{\partial a_{grav}^X}{\partial z_k} =$	$\frac{1}{\rho_i} \sum_{j=1}^N m_j \left[\left(\frac{\partial \phi_i}{\partial z_i} - \frac{\partial \phi_j}{\partial z_i} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial X} + (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial z_i \partial X} \right]$	$\frac{1}{\rho_i} m_j \left[\left(\frac{\partial \phi_i}{\partial z_j} - \frac{\partial \phi_j}{\partial z_j} \right) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial x_i} - (\phi_i - \phi_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}}{\partial x_i \partial z_i} \right]$
$\frac{\partial a_{grav}^X}{\partial \rho_k} =$	$-\frac{1}{\rho_i^2} \sum_{j=1}^N m_j (\phi_i - \phi_j) \frac{\partial \tilde{\mathbf{W}}_{ij}}{\partial X}$	0

Cuadro C.11: Gravedad según el potencial gravitatorio: Cálculo de la gravedad y sus derivadas, donde X es una variable genérica que representa las variables según las direcciones (x_i, y_i, z_i) .

C.3. Conducción térmica (κ)

Como se especifica en la sección 2.2.6 la ecuación básica que agrega el flujo de calor al código ISFAA es la descrita por *Springel, 2010* [95]. Según se escoja la implementación del modelo con la ecuación de la temperatura o de la energía se utilizará la siguiente definición:

$$\dot{T}_{cond} = \frac{1}{C_v} \cdot \dot{U}_{cond} \quad (\text{C.3})$$

En esta sección se van a desarrollar las derivadas del término de flujo de calor. Para el cálculo de las derivadas y su posterior implementación se prefiere reordenar la ecuación 2.117.

$$\left(\frac{dU_i}{dt} \right)_{Cond} = C_v \cdot \left(\frac{dT_i}{dt} \right)_{Cond} = \sum_{j=1}^N m_j \frac{1}{\rho_i \rho_j} (k_i + k_j) (T_j - T_i) \frac{\mathbf{r}_{ij}}{(r_{ij})^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \quad (\text{C.4})$$

En los diferentes modelos el valor del coeficiente de conductividad térmica κ se toma constante, una vez se ha determinado su valor medio. Esto se obtiene mediante dos métodos dependiendo

del tipo de ecuación de estado. Si la ecuación de estado es de gas ideal (por ejemplo en el problema “wall heating shock”), el valor de κ se obtiene mediante la ecuación 2.118. En ecuaciones de estado es más realistas (en nuestro caso aplicada a las enanas blancas, Sec. 5), el valor de κ se obtiene mediante el algoritmo para el cálculo de la opacidad conductiva en el caso líquido (Nandkumar y Pethick, 1984 [66]).

$\dot{U}_{cond} =$	$\sum_{j=1}^N m_j \frac{1}{\rho_i \rho_j} (k_i + k_j) (T_j - T_i) \frac{\mathbf{r}_{ij}}{r_{ij}^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$
$\frac{\partial \dot{U}_{cond}}{\partial X} =$	$\sum_{j=1}^N m_j \frac{1}{\rho_i \rho_j} (k_i + k_j) (T_j - T_i) \frac{\partial \left(\frac{\mathbf{r}_{ij}}{r_{ij}^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial X}$
$\frac{\partial \dot{U}_{cond}}{\partial E} =$	$\sum_{j=1}^N m_j \frac{1}{\rho_i \rho_j} (k_i + k_j) \frac{\partial (T_j - T_i)}{\partial E} \frac{\mathbf{r}_{ij}}{r_{ij}^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$
$\frac{\partial \dot{U}_{cond}}{\partial \rho} =$	$\sum_{j=1}^N m_j \frac{\partial (\rho_i \rho_j)^{-1}}{\partial \rho} (k_i + k_j) (T_j - T_i) \frac{\mathbf{r}_{ij}}{r_{ij}^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j)$

Cuadro C.12: Término de la conducción térmica necesario para la ecuación de la energía. Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j y E por: T_i, T_j ; U_i, U_j

$$\frac{\partial \left(\frac{\mathbf{r}_{ij}}{r_{ij}^2} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial X} = \frac{\partial \left(\frac{1}{r_{ij}^2} \right)}{\partial X} \cdot \mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) + \frac{1}{r_{ij}^2} \cdot \frac{\partial \left(\mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial X}$$

$$\begin{aligned} \frac{\partial \left(\frac{1}{r_{ij}^2} \right)}{\partial x} &= -\frac{2(x_i - x_j)}{r^4} \cdot \frac{\partial (x_i - x_j)}{\partial x} \\ \frac{\partial \left(\frac{1}{r_{ij}^2} \right)}{\partial y} &= -\frac{2(y_i - y_j)}{r^4} \cdot \frac{\partial (y_i - y_j)}{\partial y} \\ \frac{\partial \left(\frac{1}{r_{ij}^2} \right)}{\partial z} &= -\frac{2(z_i - z_j)}{r^4} \cdot \frac{\partial (z_i - z_j)}{\partial z} \end{aligned}$$

Cuadro C.13: Conducción térmica: Derivadas intermedias I. Cambiar variables genéricas X por: x_i, x_j ; y_i, y_j ; z_i, z_j

$$\mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) = (x_i - x_j) \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x} + (y_i - y_j) \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial y} + (z_i - z_j) \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial z}$$

$$\frac{\partial \left(\mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial x} = \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x} + (x_i - x_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x \partial x} + (y_i - y_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x \partial y} + (z_i - z_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x \partial z}$$

$$\frac{\partial \left(\mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial y} = \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial y} + (x_i - x_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x \partial y} + (y_i - y_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial y \partial y} + (z_i - z_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial y \partial z}$$

$$\frac{\partial \left(\mathbf{r}_{ij} \nabla_i \tilde{\mathbf{W}}_{ij}(h_i, h_j) \right)}{\partial z} = \frac{\partial \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial z} + (x_i - x_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial x \partial z} + (y_i - y_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial z \partial y} + (z_i - z_j) \frac{\partial^2 \tilde{\mathbf{W}}_{ij}(h_i, h_j)}{\partial z \partial z}$$

Cuadro C.14: Conducción térmica: Derivadas intermedias II.

C.4. Damping

El término de *damping* realiza una amortiguación en la velocidad y es utilizado para relajar los modelos cercanos al equilibrio. Una vez conseguido el equilibrio, hay que eliminarlo del esquema para realizar las simulaciones.

$\mathbf{a}_{\text{damp}} =$	$-\kappa \cdot \mathbf{v}$
$a_{\text{damp}}^k =$	$-\kappa \cdot v_k$
$\frac{\partial a_{\text{damp}}^k}{\partial v_X} =$	$-\kappa \cdot \frac{\partial v_k}{\partial v_X}$

Cuadro C.15: Damping: Amortiguación de la velocidad, donde κ es una constante y k son las variables según las direcciones (x_i, y_i, z_i) . Cambiar variables genéricas v_X por: $v_{x_i}, v_{x_j}; v_{y_i}, v_{y_j}; v_{z_i}, v_{z_j}$

Apéndice D

Ecuación de estado (EOS)

Como se especifica en la sección 2.2.7, la ecuación de estado permite definir la presión en cada partícula del modelo. La función presión depende de dos parámetros: De la densidad (ρ) y de la energía (u) o la temperatura (T), pudiendo definir su actualización a cada paso de tiempo (ecuación D.1) según sus variaciones (ecuación D.2).

$$P^{n+1} = P^n + \delta P^n \quad (\text{D.1})$$

$$\delta P = \frac{\partial P}{\partial \rho} \cdot \delta \rho + \frac{\partial P}{\partial [U//T]} \cdot \delta [U//T] \quad (\text{D.2})$$

$$C_s = \left(\frac{\partial P}{\partial \rho} \right)^{1/2} \quad (\text{D.3})$$

$$R = 8.314472 \cdot 10^7 \text{ erg} \cdot K^{-1} \cdot \text{mol}^{-1} \quad (\text{D.4})$$

$$\mu_{ion} = 1 \quad (\text{D.5})$$

$$\gamma = \frac{C_p}{C_v} = \frac{5}{3} \quad (\text{D.6})$$

$$C_v = \frac{dU}{dT} = \frac{1}{(\gamma - 1)} \frac{R}{\mu_{ion}} = \frac{3}{2} \frac{R}{\mu_{ion}} \quad (\text{D.7})$$

$$C_p = R + C_v \quad (\text{D.8})$$

En astrofísica, en general, se usa el sistema *cgs* para las unidades físicas y por tanto la constante de gases ideales usada en la ecuación de estado de gas ideal tiene un valor especificado por la ecuación D.4. En los tests de la sección 4, si no se especifica lo contrario, la composición del gas ideal se supone de Hidrógeno monoatómico, donde fijamos el peso molecular D.5 y el coeficiente

adiabático D.6. El calor específico se especifica en la ecuación D.7 a volumen constante (C_v) y D.8 a presión constante (C_p), donde se utiliza el parámetro R .

En el código, para la ecuación de estado se han implementado dos alternativas: La ecuación de estado para gases ideales o una ecuación de estado más realista que contempla los iones, las repulsiones coulombianas, la contribuciones de los electrones degenerados y la radiación. Veamos estas implementaciones en más detalle.

Gas ideal

La ecuación de estado obtiene la presión en función de la densidad y de la energía o la temperatura. En función de cual sea la variable independiente que se escoja (U, T) podremos calcular el otro parámetro según lo indicado en el cuadro D.1.

$$P(\rho, T) = \frac{R}{\mu_{ion}} \rho T \implies U(T) = \frac{1}{(\gamma - 1)} \frac{R}{\mu_{ion}} T \quad (D.9)$$

$$P(\rho, U) = (\gamma - 1) \rho U \implies T(U) = (\gamma - 1) \frac{\mu_{ion}}{R} U \quad (D.10)$$

Cuadro D.1: EOS ideal: Conversión entre energía y temperatura.

El cuadro D.2 contiene las derivadas de las variables implicadas en la ecuación de estado.

	$\partial/\partial\rho$	$\partial/\partial T$	$\partial/\partial U$	$\partial/\partial h$
P	$\frac{\partial P}{\partial \rho} = \frac{R}{\mu_{ion}} T = (\gamma - 1) U$	$\frac{\partial P}{\partial T} = \frac{R}{\mu_{ion}} \rho$	$\frac{\partial P}{\partial U} = (\gamma - 1) \rho$	$\frac{\partial P}{\partial h} = \frac{\partial \rho}{\partial h} \frac{\partial P}{\partial \rho} = -d \frac{\rho}{h} \frac{\partial P}{\partial \rho}$
T	$\frac{\partial T}{\partial \rho} = 0$	$\frac{\partial T}{\partial T} = 1$	$\frac{\partial T}{\partial U} = (\gamma - 1) \frac{\mu_{ion}}{R}$	$\frac{\partial T}{\partial h} = 0$
U	$\frac{\partial U}{\partial \rho} = 0$	$\frac{\partial U}{\partial T} = \frac{1}{(\gamma - 1)} \frac{R}{\mu_{ion}}$	$\frac{\partial U}{\partial U} = 1$	$\frac{\partial U}{\partial h} = 0$
C_s	$\frac{\partial C_s}{\partial \rho} = \frac{1}{2 C_s} \frac{\partial^2 P}{\partial \rho^2} = 0$	$\frac{\partial C_s}{\partial T} = \frac{1}{2 C_s} \frac{\partial^2 P}{\partial T \partial \rho} = \frac{\left(\frac{R}{\mu_{ion}}\right)}{2 C_s}$	$\frac{\partial C_s}{\partial U} = \frac{1}{2 C_s} \frac{\partial^2 P}{\partial U \partial \rho} = \frac{(\gamma - 1)}{2 C_s}$	$\frac{\partial C_s}{\partial h} = \frac{1}{2 C_s} \frac{\partial^2 P}{\partial h \partial \rho} = 0$

Cuadro D.2: EOS ideal: Derivadas de las ecuaciones (P presión; U energía; T temperatura; C_s velocidad del sonido local).

Ecuación realista

En el código se ha utilizado como ecuación de estado la ecuación habitualmente usada en el grupo de astronomía y astrofísica (GAA) de la UPC. Esta ecuación de estado se ha diseñado teniendo en cuenta los iones, las correcciones coulombianas, la degeneración de los electrones (*Blinnikov y colaboradores, 2011* [12]) y la radiación.

Este código supone que las variables independientes de entrada son la densidad (ρ) y la temperatura (T), obteniendo como parámetro de salida el valor de la energía).

En el programa se utiliza esta pieza de código como una caja negra, donde dadas las variables de entrada y se obtienen como salida las siguientes variables:

- P , Presión.
- $\left(\frac{\partial P}{\partial \rho}\right)_T$, Derivada de la presión respecto de la densidad a temperatura constante.
- $\left(\frac{\partial P}{\partial T}\right)_\rho$, Derivada de la presión respecto de la temperatura a densidad constante.
- U , Energía.
- $\left(\frac{\partial U}{\partial \rho}\right)_T$, Derivada de la energía respecto de la densidad a temperatura constante.
- $\left(\frac{\partial U}{\partial T}\right)_\rho$, Derivada de la energía respecto de la temperatura a densidad constante.

El código trata de forma independiente la contribución de: electrones, iones y radiación. Para calcular las contribuciones tenemos 3 subrutinas que son llamadas también de forma independiente, las variables de salida finales es el sumatorio de las subvariables obtenidas en las tres subrutinas. Como ejemplo podemos observar como se calcula la presión (ecuación D.11) y la energía (ecuación D.12) de la partícula i .

$$P_i = P_i^{(electrónica)} + P_i^{(iónica)} + P_i^{(radiación)} \quad (D.11)$$

$$U_i = U_i^{(electrónica)} + U_i^{(iónica)} + U_i^{(radiación)} \quad (D.12)$$

El cuadro D.3 contiene las derivadas de las variables implicadas en la ecuación de estado.

	$\partial/\partial\rho$	$\partial/\partial U$	$\partial/\partial T$	$\partial/\partial h$
P	$\left(\frac{\partial P}{\partial \rho}\right)_T$	---	$\left(\frac{\partial P}{\partial T}\right)_\rho$	---
U	$\left(\frac{\partial U}{\partial \rho}\right)_T$	---	$\left(\frac{\partial U}{\partial T}\right)_\rho$	---
T	--	---	---	---
C_s	$\frac{\partial C_s}{\partial \rho} = \frac{1}{2 C_s} \cdot \frac{\partial^2 P}{\partial \rho^2} \simeq 0$	$\frac{\partial C_s}{\partial U} = \frac{1}{2 C_s} \cdot \frac{\partial^2 P}{\partial U \partial \rho} \simeq 0$	$\frac{\partial C_s}{\partial T} = \frac{1}{2 C_s} \cdot \frac{\partial^2 P}{\partial T \partial \rho} \simeq 0$	$\frac{\partial C_s}{\partial T} = \frac{1}{2 C_s} \cdot \frac{\partial^2 P}{\partial h \partial \rho} \simeq 0$

Cuadro D.3: EOS realista: Derivadas de las variables (P presión; U energía; T temperatura; C_s velocidad del sonido local), obtenidas por las subrutinas. Donde las derivadas de la C_s son aproximadas a cero para simplificar los cálculos.

La energía como variable independiente. La energía depende de la densidad y la temperatura, su variación se describe en la ecuación D.13. Si se toma $\rho = cte$ se elimina el primer sumando y como $C_v = \partial U / \partial T$ se tiene la ecuación D.15 que es el factor de conversión entre la variación de la energía δU y la variación de la temperatura δT .

$$\delta U = \left(\frac{\partial U}{\partial \rho}\right)_T \cdot \delta \rho + \left(\frac{\partial U}{\partial T}\right)_\rho \cdot \delta T \quad (D.13)$$

$$\delta T = \left(\frac{\partial T}{\partial \rho}\right)_U \cdot \delta \rho + \left(\frac{\partial T}{\partial U}\right)_\rho \cdot \delta U \quad (D.14)$$

$$\delta U = C_v \cdot \delta T \quad (D.15)$$

Usando la energía en las ecuaciones implícitas aparecerán una serie de derivadas que no se obtienen en las subrutinas (ecuación D.16) y que será necesario calcular.

$$1) \left(\frac{\partial P}{\partial \rho}\right)_U, \quad 2) \left(\frac{\partial T}{\partial \rho}\right)_U \quad ; \quad 3) \left(\frac{\partial P}{\partial U}\right)_\rho, \quad 4) \left(\frac{\partial T}{\partial U}\right)_\rho \quad (D.16)$$

Para obtener 1) se hace de forma directa porque la temperatura T es una función de la energía $T(U)$. Se puede desarrollar el término en la ecuación D.17. Podemos observar que nos

falta el factor $\left(\frac{\partial T}{\partial \rho}\right)_U$ que se corresponde con la derivada 2).

$$\left(\frac{\partial P}{\partial \rho}\right)_U = \left(\frac{\partial P}{\partial \rho}\right)_T + \left(\frac{\partial P}{\partial T}\right)_\rho \cdot \left(\frac{\partial T}{\partial \rho}\right)_U = \left(\frac{\partial P}{\partial \rho}\right)_T - \left(\frac{\partial P}{\partial T}\right)_\rho \cdot \frac{\left(\frac{\partial U}{\partial \rho}\right)_T}{\left(\frac{\partial U}{\partial T}\right)_\rho} \quad (\text{D.17})$$

Para obtener 2) se aplica la regla de la cadena, cambiando los términos constantes en el proceso, y se puede desarrollar el término en la ecuación D.18.

$$\left(\frac{\partial T}{\partial \rho}\right)_U = - \left(\frac{\partial T}{\partial U}\right)_\rho \cdot \left(\frac{\partial U}{\partial \rho}\right)_T = - \left(\frac{\partial U}{\partial T}\right)_\rho^{-1} \cdot \left(\frac{\partial U}{\partial \rho}\right)_T = - \frac{\left(\frac{\partial U}{\partial \rho}\right)_T}{\left(\frac{\partial U}{\partial T}\right)_\rho} \quad (\text{D.18})$$

Para obtener 3) se aplica la regla de la cadena y se puede desarrollar el término en la ecuación D.19. En este caso tenemos todos los factores ya que son salidas de la eos realista.

$$\left(\frac{\partial P}{\partial U}\right)_\rho = \left(\frac{\partial P}{\partial T}\right)_\rho \cdot \left(\frac{\partial T}{\partial U}\right)_\rho = \left(\frac{\partial P}{\partial T}\right)_\rho \cdot \left(\frac{\partial U}{\partial T}\right)_\rho^{-1} = \frac{\left(\frac{\partial P}{\partial T}\right)_\rho}{\left(\frac{\partial U}{\partial T}\right)_\rho} \quad (\text{D.19})$$

Para obtener 4) invertimos la fracción y se puede desarrollar el término en la ecuación D.20.

$$\left(\frac{\partial T}{\partial U}\right)_\rho = \left(\frac{\partial U}{\partial T}\right)_\rho^{-1} = (C_v)^{-1} = \frac{1}{C_v} \quad (\text{D.20})$$

Cálculo de la temperatura mediante el método *Newton-Raphson* Cuando se escoge como variable independiente la energía, como la subrutina solo acepta como entrada densidad y temperatura, tenemos el problema de la determinación del parámetro temperatura que corresponde a la energía de la partícula del sistema.

Como se ha visto anteriormente en la ecuación D.14 la temperatura depende de la densidad y la energía. Para determinar este parámetro, se coge la energía de salida del código, que se denominara U_{EOS} y se comparará con el valor en SPH , que se denominará U_{SPH} . Mediante un método *Newton-Raphson* se irá refinando de forma iterativa el valor de la temperatura hasta que coincidan las energías con un cierto margen de error aceptado, $|U_{EOS} - U_{SPH}| \leq \varepsilon_U$.

El método *Newton-Raphson*, para una ecuación $f(x)$ con una variable independiente x se define según la ecuación D.21, descrita en la ecuación 2.161.

$$x_i^{n+1} = x_i^n - \frac{f(x_i)}{f'(x_i)} \quad (\text{D.21})$$

Traduciendo el método al esquema, se tiene que escoger una T_o (basada en la T estable de paso de tiempo anterior) y la ρ real que corresponda como parámetros de entrada en $EOS(\rho_{SPH}, T_o)$. Como resultado se obtiene una presión P y una energía U_{EOS} que se comparará con la de la partícula U_{SPH} . Si difieren en un margen mayor que un cierto valor ε_U , se retoca T_o con el *Newton-Raphson* y se reintroduce en la subrutina $EOS(T'_o, \rho)$ a densidad constante, repitiéndose el proceso hasta encontrar el valor correcto.

Según la relación entre variación de energía y temperatura (ecuación D.15), despejamos la variación de temperatura y igualamos a cero, según indica el método *Newton-Raphson*, obteniendo la ecuación D.22

$$\delta T + \frac{\delta U}{C_v} = 0 \quad (D.22)$$

Para cambiar la variable independiente x de la ecuación D.21, por la variable independiente T , se necesita desarrollar los términos de la ecuación D.22, obteniendo lo indicado en el cuadro D.4

$$\delta T = \frac{dT}{dt} = \frac{\Delta T}{\Delta t} = \frac{T - T_o}{\Delta t} \quad (D.23)$$

$$\delta U = \frac{dU}{dt} = \frac{\Delta U}{\Delta t} = \frac{U - U_o}{\Delta t} \quad (D.24)$$

$$C_v = \frac{dU}{dT} \quad (D.25)$$

Cuadro D.4: Factores de relación entre energía (U) y temperatura (T).

Sustituyendo los valores, obtenemos la ecuación del refinamiento de la temperatura D.26, donde podemos aplicar la condición de convergencia D.27 que compara con la diferencia relativa entre las energías.

$$T = T_o - \frac{U - U_o}{C_v} \quad (D.26)$$

$$\varepsilon_U \geq \frac{|U_{EOS} - U_{SPH}|}{U_{SPH}} \quad (D.27)$$

Donde el error ε_U puede estar en el rango $[10^{-8}, 10^{-4}]$.

Apéndice E

Análisis y convergencia en las ecuaciones

Definiciones para analizar las ecuaciones

Para analizar las ecuaciones y su convergencia necesitamos establecer una serie de definiciones descritas en detalle en *Menéndez, 2010* [56].

Error Existen tres tipos principales de error: inherentes, asociados a los datos de entrada (medición, secuencia finita de dígitos, retroalimentación de la entrada, ...); truncamiento, asociados a truncar un proceso matemáticamente infinito (evaluación de series con sumas finitas, derivación por aproximaciones en diferencias finitas, ...); redondeo, asociado a la conservación de un número finito de dígitos (precisión de la máquina). El estudio de la propagación de los errores permite acotar el error total.

Estabilidad Un método es estable si es poco sensible a pequeñas variaciones en los datos de entrada. Se mide en la unidad mínima que puede manejar la máquina que estamos usando y sus causas pueden ser dos: inherente al problema, porque el problema este mal acondicionado o porque sea matemáticamente inestable; inherente al método, porque el método este mal construido o porque sea numéricamente inestable. La inestabilidad no se manifiesta necesariamente para cualquier conjunto de valores de entrada. En el caso de métodos numéricamente inestables (es decir problema bien condicionado), es necesario reemplazarlo por otro matemáticamente equivalente que sea estable o adaptarlo para que el original sea estable. En una ecuación en diferencias de dos niveles en tiempo o espacio (primer orden) es condición suficiente para garantizar

la estabilidad si tiene solo una variable dependiente. Si un algoritmo iterativo (véase sección 2.3.2) es inestable, los errores crecen exponencialmente en valor absoluto (de forma monótona u oscilatoria), en general, esto no sucede con los métodos directos.

Estabilidad en sistemas de ecuaciones no lineales La no linealidad de un problema complica el análisis de estabilidad, porque el análisis debe hacerse sobre la ecuación en diferencias original linealizada. En la literatura se pueden encontrar las descripciones para linealizar ecuaciones: Menéndez, 2010 [56], Dahlquist y Björck, 2003 [23], etc. En problemas no lineales una fuente importante de inestabilidad radica en la transferencia de energía. No existe un mecanismo eficaz de disipación y por tanto se acumula energía en las partículas. Esto otorga un aspecto ruidoso a la solución numérica, que puede indicar el inicio de un proceso inestable. Si el problema que se trata tiene, escalas temporales cortas, un mecanismo de disipación mediante un término difusivo puede evitar esa inestabilidad. Pero, si se tienen números de Reynolds altos la disipación puede resultar insuficiente y es necesario agregar una viscosidad artificial con un término difusivo extra para eliminar la energía a escalas pequeñas o utilizar esquemas numéricos disipativos, es decir, que introducen difusión numérica. Es necesario verificar que la estabilidad numérica no se logre a costa de una inaceptable pérdida de precisión.

Consistencia Al aproximar una ecuación diferencial discretizándola en una ecuación de diferencias se comete un error de truncamiento. Se denomina error de discretización a la diferencia entre la solución aproximada de la ecuación de diferencias y la ecuación diferencial o solución analítica. Dada una variable ν , una solución aproximada ν' es consistente si se acerca a la solución analítica una distancia menor o igual que un cierto error de discretización ε que se acepta para el método ($\nu - \varepsilon \leq \nu' \leq \nu + \varepsilon$). Es decir que un esquema numérico es consistente con una ecuación diferencial, si el error de discretización tiende a anularse cuando disminuyen continuamente los intervalos de discretización es decir, $\varepsilon_j^n \rightarrow 0$, cuando $\Delta t \rightarrow 0, \Delta x \rightarrow 0$. La condición de consistencia es necesaria, aunque no suficiente, para que la solución numérica converja a la solución analítica cuando se afina la malla de cálculo.

Criterio de Courant El criterio de Courant (Courant y colaboradores, 1967 [22]) establece que el dominio de influencia (respecto de un punto) de la solución analítica debe de estar contenido en el de la solución numérica, tal cual se muestra en la figura E.1. La violación de este criterio produce inestabilidades numéricas de tipo oscilatorio. De este principio general se deduce la *Condición de Courant* que limita el paso de tiempo en los métodos explícitos para garantizar la estabilidad numérica.

$$\Delta t \leq \frac{\Delta x}{U} \quad (\text{E.1})$$

donde Δt es el paso de tiempo, Δx es la variación en posición en 1D y U representa la velocidad del sonido que representa la longitud máxima de una celda de resolución.

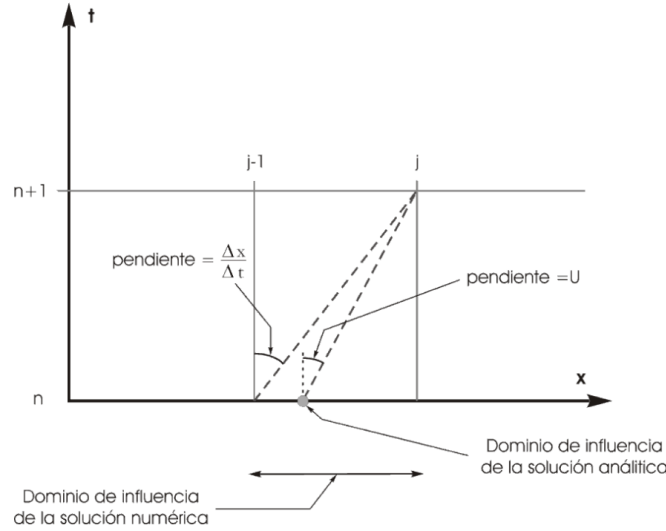


Figura E.1: Criterio de Courant para la estabilidad de una ecuación en diferencias. En SPH, Δx se relaciona con la longitud de suavizado y U se sustituye por la velocidad del sonido local (C_s).

Convergencia Un método es convergente si es estable y además es consistente, es decir, la solución aproximada tiende a la solución analítica al refinar la malla de cálculo, tal que, $\nu_j^n \rightarrow u(x_j, t^n)$, cuando $\Delta t \rightarrow 0, \Delta x \rightarrow 0$ y en el límite los índices j y n varían ($j, n \rightarrow \infty$) mientras que x_j y t^n permanecen fijos (por ejemplo, $x_j = j\Delta x$ y $t^n = n\Delta t$). Cuando se analiza la convergencia, hay que observar que, al afinar la malla de cálculo el error de truncamiento disminuye, mientras los errores de redondeo permanecen bajo control. La consistencia es necesaria para que el error de truncamiento disminuya al refinar la malla. Además, hay que comprobar que el esquema de variables permanece estable (*Richtmyer y Morton, 1967* [77]).

Orden de precisión La precisión de una solución numérica ν_j^n se mide por su diferencia con la solución analítica $\nu(x_j, t^n)$ para cada nodo (j, n) de la discretización. En un esquema numérico son las potencias dominantes de los intervalos de discretización correspondientes a cada una de las variables independientes, según su error de discretización. Controlar la precisión significa cuantificar la diferencia (error de truncamiento) respecto de la solución analítica. El orden de precisión mide con que ritmo se anula el error de discretización cuando se afina la malla.

Criterios de convergencia

Como se especifica en la sección 2.3.4 para garantizar la convergencia del método es necesario establecer una serie de criterios que afectan a las ecuaciones del esquema implícito.

Convergencia en las correcciones

Este criterio es uno de los criterios estándar que mas se utilizan para determinar la convergencia de un esquema *Newton-Raphson*. Se basa en comprobar las correcciones normalizadas obtenidas en el último refinamiento.

$$\left| \frac{\delta \chi_i}{\chi_i} \right| \leq \varepsilon \quad (\text{E.2})$$

si todas las correcciones de las variables del esquema se sitúan por debajo de un valor $\varepsilon = 10^{-4}$, la convergencia del sistema se da por correcta.

Existe la posibilidad de que cuando las variables independientes son de tipos diferentes (como por ejemplo energía interna y velocidad), controlar solo uno o varios tipos de estas variables.

También es posible permitir un porcentaje de variables que incumplan este criterio, para evitar que determinen los pasos de tiempo las zonas de la simulación donde los valores de las variables son practicamente nulos y sus correcciones igual, ya que se se sitúan en su rango de valores.

Convergencia en las ecuaciones

Este criterio es mas restrictivo que el estándar de los métodos *Newton-Raphson* pero garantiza la estabilidad del esquema. Para ello se descompone cada una de las ecuaciones iG_j en sumas, de tal modo que ${}^iG_j = ({}^iG_j)_1 + ({}^iG_j)_2 + \dots + ({}^iG_j)_n$, donde i representa cada partícula, j cada ecuación y n cada uno de los sumandos. Es decir, una vez actualizadas las variables, aplicamos el siguiente criterio E.3, donde ε se suele tomar en el rango $[10^{-4}, 10^{-1}]$.

$$\frac{|{}^iG_j|}{\sqrt{({}^iG_j)_1^2 + ({}^iG_j)_2^2 + \dots + ({}^iG_j)_n^2}} \leq \varepsilon \quad (\text{E.3})$$

Pasemos a ver los criterios de convergencia de cada una de las ecuaciones que pueden intervenir en el esquema.

Ecuación del momento Dadas las siguientes ecuaciones [E.4](#), [E.5](#) y [E.6](#).

$$G_{ax} = \frac{v_x - v_x^{(n-1)}}{\Delta t} - (1-\theta)a_x^{(n-1)} - \theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial x} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial x} \right) + a_{visco}^x + a_{grav}^x \right] \quad (\text{E.4})$$

$$G_{ay} = \frac{v_y - v_y^{(n-1)}}{\Delta t} - (1-\theta)a_y^{(n-1)} - \theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial y} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial y} \right) + a_{visco}^y + a_{grav}^y \right] \quad (\text{E.5})$$

$$G_{az} = \frac{v_z - v_z^{(n-1)}}{\Delta t} - (1-\theta)a_z^{(n-1)} - \theta \left[-\sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial z} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial z} \right) + a_{visco}^z + a_{grav}^z \right] \quad (\text{E.6})$$

Se puede descomponer G_{a_k} en sumandos, suponiendo $k = x, y, z$, se obtienen las tres ecuaciones [E.7](#) a comprobar.

- $(G_{a_k})_1 = \frac{v_k - v_k^{n-1}}{\Delta t}$
- $(G_{a_k})_2 = -(1-\theta) a_k^{n-1}$
- $(G_{a_k})_3 = \theta \sum_{j=1}^N m_j \left(\frac{P_i}{\Omega_i \rho_i^2} \frac{\partial W_{ij}(h_i)}{\partial k} + \frac{P_j}{\Omega_j \rho_j^2} \frac{\partial W_{ij}(h_j)}{\partial k} \right)$
- $(G_{a_k})_4 = -\theta a_{visco}^k$
- $(G_{a_k})_5 = -\theta a_{grav}^k$

$$\frac{|G_{a_k}|}{\sqrt{(G_{a_k})_1^2 + (G_{a_k})_2^2 + (G_{a_k})_3^2 + (G_{a_k})_4^2 + (G_{a_k})_5^2}} \leq \varepsilon_a \quad (\text{E.7})$$

Ecuación de la energía Dada la siguiente ecuación [E.8](#).

$$G_{\dot{U}} = \frac{U - U^{n-1}}{\Delta t} - (1-\theta)\dot{U}^{n-1} - \theta \left[\frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{U}_{visco} + \dot{U}_{cond} \right] \quad (\text{E.8})$$

Se puede descomponer G_U en sumandos y se obtiene la ecuación [E.9](#) a comprobar.

- $(G_U)_1 = \frac{U - U^{n-1}}{\Delta t}$
- $(G_U)_2 = -(1-\theta) U^{n-1}$

- $(G_U)_3 = -\theta \frac{P_i}{\Omega_i \rho_i^2} \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i)$
- $(G_U)_4 = -\theta \dot{U}_{visco}$
- $(G_U)_5 = -\theta \dot{U}_{cond}$

$$\frac{|G_U|}{\sqrt{(G_U)_1^2 + (G_U)_2^2 + (G_U)_3^2 + (G_U)_4^2 + (G_U)_5^2 + (G_U)_6^2}} \leq \varepsilon_U \quad (\text{E.9})$$

Ecuación de la temperatura Dada la siguiente ecuación [E.10](#).

$$G_{\dot{T}} = \frac{T - T^{n-1}}{\Delta t} - (1-\theta) \dot{T}^{n-1} - \theta \left[\frac{1}{\Omega_i \rho_i^2} \cdot \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \cdot \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i) + \dot{T}_{visco} + \dot{T}_{cond} \right] \quad (\text{E.10})$$

Se puede descomponer G_T en sumandos y se obtiene la ecuación [E.11](#) a comprobar.

- $(G_T)_1 = \frac{T - T^{n-1}}{\Delta t}$
- $(G_T)_2 = -(1-\theta) \dot{T}^{n-1}$
- $(G_T)_3 = -\theta \frac{1}{\Omega_i \rho_i^2} \cdot \frac{\partial P_i}{\partial T_i} \frac{T_i}{C_v} \cdot \sum_{j=1}^N m_j (v_i - v_j) \nabla_i \mathbf{W}_{ij}(h_i)$
- $(G_T)_4 = -\theta \dot{T}_{visco}$
- $(G_T)_5 = -\theta \dot{T}_{cond}$

$$\frac{|G_T|}{\sqrt{(G_T)_1^2 + (G_T)_2^2 + (G_T)_3^2 + (G_T)_4^2 + (G_T)_5^2 + (G_T)_6^2}} \leq \varepsilon_T \quad (\text{E.11})$$

Ecuación del movimiento Dadas las siguientes ecuaciones [E.12](#), [E.13](#) y [E.14](#).

$$G_{v_x} = \frac{x - x^{n-1}}{\Delta t} - (1-\theta) v_x^{n-1} - \theta [v_x] \quad (\text{E.12})$$

$$G_{v_y} = \frac{y - y^{n-1}}{\Delta t} - (1-\theta) v_y^{n-1} - \theta [v_y] \quad (\text{E.13})$$

$$G_{v_z} = \frac{z - z^{n-1}}{\Delta t} - (1-\theta) v_z^{n-1} - \theta [v_z] \quad (\text{E.14})$$

Se puede descomponer G_{v_k} en sumandos, suponiendo $k = x, y, z$, se obtienen las tres ecuaciones [E.15](#) a comprobar.

- $(G_{v_k})_1 = \frac{k - k^{n-1}}{\Delta t}$

- $(G_{v_k})_2 = -(1 - \theta) v_k^{n-1}$
- $(G_{v_k})_3 = -\theta v_k$

$$\frac{|G_{v_k}|}{\sqrt{(G_{v_k})_1^2 + (G_{v_k})_2^2 + (G_{v_k})_3^2}} \leq \varepsilon_v \quad (\text{E.15})$$

Ecuación de la densidad Dada la siguiente ecuación [E.16](#).

$$G_{\rho} = \rho - \theta \left[\frac{1}{\theta} \sum_{j=1}^N m_j W_{ij}(h_i) \right] \quad (\text{E.16})$$

Se puede descomponer G_{ρ} en sumandos y se obtiene la ecuación [E.17](#) a comprobar.

- $(G_{\rho})_1 = \rho$
- $(G_{\rho})_2 = -\sum_{j=1}^N m_j W_{ij}(h_i)$

$$\frac{|G_{\rho}|}{\sqrt{(G_{\rho})_1^2 + (G_{\rho})_2^2}} \leq \varepsilon_{\rho} \quad (\text{E.17})$$

Ecuación de la continuidad Dada la siguiente ecuación [E.18](#).

$$G_{\dot{\rho}} = \frac{\rho - \rho^{n-1}}{\Delta t} - (1 - \theta) \dot{\rho}^{n-1} - \theta \left[\sum_{j=1}^N m_j (v_i - v_j) \nabla_{\mathbf{i}} \mathbf{W}_{\mathbf{ij}}(h_i) \right] \quad (\text{E.18})$$

Se puede descomponer $G_{\dot{\rho}}$ en sumandos y se obtiene la ecuación [E.19](#) a comprobar.

- $(G_{\dot{\rho}})_1 = \frac{\rho - \rho^{n-1}}{\Delta t}$
- $(G_{\dot{\rho}})_2 = -(1 - \theta) \dot{\rho}^{n-1}$
- $(G_{\dot{\rho}})_3 = -\theta \sum_{j=1}^N m_j (v_i - v_j) \nabla_{\mathbf{i}} \mathbf{W}_{\mathbf{ij}}(h_i)$

$$\frac{|G_{\dot{\rho}}|}{\sqrt{(G_{\dot{\rho}})_1^2 + (G_{\dot{\rho}})_2^2 + (G_{\dot{\rho}})_3^2}} \leq \varepsilon_{\dot{\rho}} \quad (\text{E.19})$$

Ecuación de la distancia de interpolacion Dada la siguiente ecuación [E.20](#).

$$G_h = h - \theta \left[\frac{1}{\theta} \left(\frac{K}{\rho} \right)^{1/d} \right] \quad (\text{E.20})$$

Se puede descomponer G_h en sumandos y se obtiene la ecuación [E.21](#) a comprobar.

$$\begin{aligned} \blacksquare (G_h)_1 &= h \\ \blacksquare (G_h)_2 &= - \left(\frac{K}{\rho} \right)^{1/d} \end{aligned}$$

$$\frac{|G_h|}{\sqrt{(G_h)_1^2 + (G_h)_2^2}} \leq \varepsilon_h \quad (\text{E.21})$$

Apéndice F

ISFAA y PANGEA

El código ISFAA inició su desarrollo a principios de 2.009 con el objetivo de conseguir un código SPH implícito operativo. El tratamiento de modelos implícitos tienen un alto coste computacional y por tanto es necesario el uso de supercomputadores para hacer factible la ejecución del código paralelo.

ISFAA

Durante estos años de desarrollo se han creado varias versiones del código, en diferentes lenguajes de programación:

TimeLine

- ISFAA 1.0 (2010): Primera versión operativa, escrita en Fortran95, podía realizar test básicos con viscosidad artificial, pero sin gravedad.
- ISFAA 2.5 (2012): Versión completa escrita en Fortran95, realiza test con viscosidad artificial, gravedad y difusión de energía. Se aplica a enanas blancas.
- ISFAA 3.0 (2013): Versión simple con viscosidad, rediseñada desde cero en Fortran95, para subsanar problemas de convergencia de las versiones anteriores.
- ISFAA 4.0 (2015): Versión completa escrita en C++, corrige los problemas de convergencia e integra viscosidad artificial y gravedad. Experimenta con test de turbulencia, modelos politrópicos de masa solar y modelos sin velocidad. Incluye una base de datos para poder ejecutar múltiples modelos sin recompilar el binario.

ISFAA C++ v.4.0

La versión actual del código esta pensada para que sea la base de múltiples ampliaciones del código. Comentemos varias de sus características para interesantes:

- Esta diseñada en UML y de forma modular, aprovechando las herramientas y patrones utilizados habitualmente en ingeniería del software.
- Utiliza el sistema de superclases y subclasses para implementar diferentes versiones del kernel, ecuaciones del esquema, gravedad, etc.
- Integra una base de datos de tipo SQLite que permite hacer tuning del código, en tiempo de ejecución, para especificar las preferencias del usuario en la simulación que va a llevar a cabo. Además, sin recompilar el código, permite al usuario ejecutar diferentes modelos con opciones diversas.
- Modificando la BD mientras el programa esta corriendo en el sistema, es capaz de modificar su ejecución. Esto habitualmente se utiliza para obtener información adicional en la ejecución de la simulación actual.
- Permite el tratamiento simple de ficheros externos a la aplicación. Se especifican las columnas de las variables del fichero externo y pueden ser usadas como entrada del sistema.
- Genera salidas del modelo en cada iteración, así como información de valores asociados a las partículas. Estos ficheros contienen una leyenda y se ofrece una serie de scripts, en bash y gnuplot, que generan imágenes y vídeos interesantes para analizar los resultados.

Herramienta de soporte

Paralelamente al desarrollo del código de la tesis. En el año 2014 inicié el desarrollo de una interfaz gráfica, escrita en Qt5, para analizar las salidas de las simulaciones de ISFAA o de cualquier otro programa. Este proyecto continúa y amplía una pequeña aplicación que desarrollé en 2007, utilizando Qt3 y OpenGL. Esta aplicación integra una base de datos que ha sido el germen del diseño del código ISFAA 4.0. Las características principales son las siguientes:

- Ventana de navegación por el disco del sistema.
- Ventana de terminal del sistema para comandos y ejecuciones del programa.
- Editor de código con resaltado de las palabras reservadas del lenguaje.

- Apertura de documentos llamando a aplicaciones externas.
- Base de datos SQLite para especificar las columnas de los ficheros de entrada.
- Ventana 3D en OpenGL para la navegación, el tratamiento y el análisis de modelos de simulación.
- Ventana 2D en Qt para obtener información del modelo de simulación.
- Panel de resultados obtenidos de los análisis de las ventanas anteriores.

El proyecto se encuentra en desarrollo y se continuará durante el 2016.

PANGEA: La máquina de simulación.

Las bibliotecas algorítmicas escogidas (PARDISO), utilizan OpenMP para su ejecución, y por tanto era necesario una máquina multiprocesador con memoria compartida (entre todos sus núcleos) o un sistema que, de forma transparente al programa, le haga ver al sistema una pseudo-memoria compartida.

Dada la dificultad de acceso a los supercomputadores para programas en desarrollo, en vez de en producción, se busco financiación para comprar un servidor dedicado al desarrollo del nuevo código. Gracias al proyecto ESF EUROCORES Program Eurogenesis a través del proyecto MICINN EUI2009-04167, a finales de 2.009 se ofreció la posibilidad de dedicar 5.000€ a la compra de un servidor para el proyecto.

Características necesarias Durante un mes valore las diferentes posibilidades de composición de la máquina con dos premisas:

1. La matriz a tratar es enorme a pesar de ser muy dispersa. Su tamaño depende del número de partículas del modelo, número de ecuaciones por partícula y número de vecinos por partícula. El tamaño mínimo de la RAM que se debía planificar era de 12Gb para conseguir trabajar con modelos al menos 10.000 partículas, 8 ecuaciones y 100 vecinos
2. Una vez cabe la matriz en memoria, el problema es exclusivamente de cálculo. Por tanto, había que maximizar el número de procesadores de la máquina, con un mínimo de 4.

Elección de la arquitectura Primeramente era necesario escoger la arquitectura de la máquina, ya que esta parte condiciona el resto de componentes del servidor. Observando comparativas vi claramente que la elección entre Intel y AMD no era tal, ya que para procesadores de características muy similares Intel siempre salía victorioso, con rendimientos superiores al 20 % por tanto me centre en este tipo de procesadores. Las líneas tecnológicas candentes en aquel momento eran la primera generación de i3, i5, i7 para ordenadores domésticos y la línea Intel Xeon W o X 5500 para el uso profesional (gamma correspondiente al i7 en uso domestico). Existía una línea superior de muy alto rendimiento que ya no se valoró por su elevado coste.

Escoger Intel i7 o Xeon tenía varias ventajas:

- La tecnología Intel “HyperThreading” o HT, donde se aprovecha el hardware interno del procesador para usar 2 hilos de ejecución por núcleo, de tal forma que una pastilla de 4 núcleos pueden funcionar como 8 hilos de ejecución consiguiendo una efectividad sobre el 70-80 % de 8 núcleos únicos.
- La tecnología Intel “TurboBoost”, que permite trabajar a cada núcleo sobre un 120 % de su capacidad de forma puntual, asumiendo picos de trabajo. Esta tecnología es buena pero no esencial ya que ISFAA produce una carga constante de trabajo y no hay picos.
- Estos procesadores pueden trabajar con memoria DDR3 en formato “triple-channel” con tres bancos de memoria con tres “slots”, de tal forma que pueden acceder a 9 dimms de memoria. Esto permite aumentar la capacidad teórica de memoria sobre los procesadores anteriores que tenían tecnología “double-channel” con dos bancos de memoria y dos “slots”, es decir 4 dimms de memoria. En ese momento existían placas de 4GB y 8GB, siendo las últimas 4 veces más caras y por tanto siendo descartadas por su coste.
- En el mercado existen placas duales con la posibilidad de montar dos pastillas de núcleos (Intel i7 o Xeon) y pudiendo acceder de forma simultanea al doble de canales de memoria y colocar el doble de dimms, es decir, 18 “slots” de memoria.
- En aquel momento ya se vislumbraba el potencial de las GPUs, y existían placas con la posibilidad de colocar hasta 4 slots PCI-E 16x para tarjetas gráficas (GPUs) pudiendo trabajar todas ellas de forma simultánea.

Servidor Objetivo Las características que se buscaban estaban ya definidas:

- Placa base dual con 2 Procesadores Intel Xeon de la serie W ó X 5500 de 4 núcleos cada uno (8 hilos), para un total de 16 hilos de ejecución.

- Un mínimo de 24GB Memoria DDR3 no-ECC a 1.333 Mhz. Es decir, 2 packs de 3 barras de DDR3 (por el triple-channel), haciendo un total de 12 barras de 4GB.
- Un mínimo de dos “slots” PCI-E 16x para poder portar el código a GPUs en el futuro, siendo deseable 4 “slots” PCI-E 16x.

Proceso de compra Primeramente valoré la posibilidad de comprar un servidor “comercial”, montado por una de las empresas que cubre este sector de servidores (IBM, HP, Dell, etc.). El presupuesto para la máquina objetivo se iba a los 12.000€, más del doble de la cantidad disponible, por tanto esta opción se descarto.

La siguiente opción era montar un servidor “a la carta”, teniendo el riesgo intrínseco y real de que al agrupar las piezas la máquina no las admitiese en conjunto. Después de más de un mes de consultar manuales, comparativas y foros, me decidí por la siguiente configuración.

- Placa base dual ASUS Z8PE-D18, con 18 Dimms de memoria RAM DDR3 y 4 PCI-E 16x, aunque por su colocación y características solo dos PCI-E son usables simultáneamente para la ejecución de GPUs. Coste aproximado 500€.
- 2 procesadores de 4 núcleos Intel Xeon W5580 a 3,2Ghz, el tope de gama en ese momento, pudiendo ejecutar 16 hilos. Coste aproximado 2.400€.
- 48GB de RAM Kingstom DDR3 no-ECC a 1.333 Mhz. Repartidos en 4 packs de 3 barras de 4GB, las económicamente mas viables Coste aproximado 1.400€.
- Nvidia Gforce 9500 GT, con 32 núcleos CUDA para hacer las primeras pruebas de portabilidad. Coste aproximado 100€. Hoy en día (tres años después) por el mismo dinero una tarjeta Nvidia Gforce GT 640 con 384 núcleos.
- Resto de componentes: 2 ventiladores de CPU, Disco duro 1.5TB, cables, fuente de alimentación especial 1200W y caja básica de servidor. Coste aproximado 600€.

Después de la petición de presupuesto y posterior negociación se acordó la compra por una cantidad cercana a 5.000€ en la sucursal de *Pista Cero* situada en la facultad de Arquitectura técnica en Barcelona en el Campus Sud de la UPC. El pago lo coordinó la secretaria del departamento de Física a un plazo de 90 días.

Servidor en producción Cuando llego la máquina el trabajo previo dio sus frutos, y todo funciono a la primera. En un día el equipo estaba montado y en funcionamiento con un sistema operativo Ubuntu Linux.

Después de una semana de pruebas y reinstalaciones para probar el rendimiento, se instaló el sistema Ubuntu Desktop 8.04 LTS 64bits, se le asignó al servidor el nombre de red PANGAEA, se colocó en un cuarto climatizado denominado Cluster de Astrofísica en la EUETIB de la UPC y se puso disponible para el acceso a través de internet.

De este modo el sistema entró en producción al inicio de 2010, siendo actualizado a Ubuntu Desktop 10.04 LTS 64bits cuando la versión fue liberada. Desde entonces ha sido usado de forma exclusiva e ininterrumpida en el desarrollo de código ISFAA hasta las fechas de escritura de esta tesis, es decir durante los casi tres años posteriores.



Figura F.1: Pangea: Servidor, con la tapa lateral abierta y recibiendo la refrigeración directa desde el aire acondicionado.



Figura F.2: Pangea: Conexiones, donde se ve el switch de red general (el de color blanco), dos switches de red pertenecientes al cluster de astrofísica y dos switches KVM de teclado y ratón. Pangea usa conexión directa a través del switch de red general y se encuentra conectado a la boca 16 del Switch KVM 1.

Apéndice G

Aceleradoras gráficas y coprocesadores

Hoy en día un ordenador personal esta compuesto de varias piezas claves:

- Procesador multinúcleo (CPUs): Con sus registros y memoria cache separada en niveles (L1,L2,...).
- Memoria principal (RAM): Que puede ser de muchos tipos, velocidades y anchos de banda(DDR-2,DDR-3,...), dependiendo de la placa base y el procesador.
- Tarjeta gráfica (GPUs): En 1 o varios “slots” PCI-E 16x con memoria dedicada habitualmente entre 1Gb-4Gb (DDR-3,DDR-5,...), que pueden contener desde decenas hasta miles de núcleos.

En la última década los procesadores gráficos han experimentando una gran evolución en su potencia de cálculo, además el aumento de las velocidades del los procesadores multinúcleo convencionales y el tamaño de las memorias están llegado a un límite físico de la tecnología. El desarrollo de procesadores con mayores capacidades depende de las posibilidades de miniaturización en circuitos integrados, que cada vez se encuentra con mayores dificultades técnicas relacionadas con la escala. La dificultad para aumentar la frecuencia del reloj, como la lentitud de transferencia de información por el bus entre memoria, procesador y disco motiva el uso de la computación paralela. Hasta épocas recientes el uso de supercomputación intensiva estaba restringido al uso de clusters carísimos en el campo de HPC (*High Performance Computing*), pero esta situación ha cambiado con la irrupción en el mercado de primeramente las CPU's multinúcleo y posteriormente, sobre el año 2.000, las GPUs (tarjetas de proceso gráfico para ordenadores de escritorio).

Las tarjetas gráficas, inicialmente orientadas al uso en la industria del cine o al tratamiento de polígonos en videojuegos, por su gran capacidad de computo paralelo han ido migrando su uso hacia la computación. El resultado es que ha día de hoy las GPU son unidades de proceso altamente segmentadas y compuestas por cientos de procesadores optimizados para el tratamiento de la información en paralelo y la realización de operaciones aritméticas de forma intensiva en coma flotante. Es por ello que el procesamiento paralelo cada vez es mas accesible al usuario personal. Hoy en día cualquier programador puede programar en paralelo y ejecutar a un costo asequible. De hecho existen placas base que admiten el uso de hasta 6 tarjetas de GPUs de forma simultánea, siendo el único requisito adicional disponer de varias fuentes de alimentación que suministren más potencia al sistema.

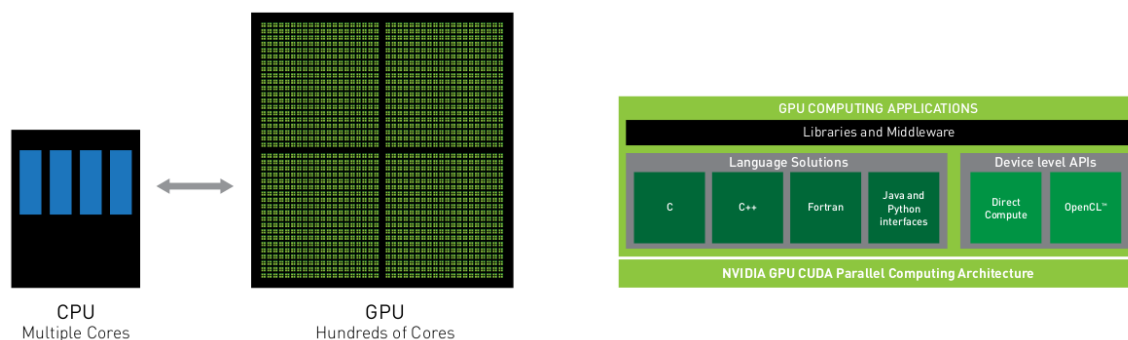


Figura G.1: Comparativa con aplicaciones entre CPU y GPU (<http://www.nvidia.es/object/gpu-computing-es.html>).

GPUs

Tarjetas Nvidia Tesla (CUDA) Los fabricantes principales de GPUs han desarrollado tecnologías para poder aprovechar la capacidad de las tarjetas y orientarlas al cálculo, siendo la empresa puntera en el campo Nvidia. El fabricante Nvidia, ha desarrollado el lenguaje de programación CUDA para facilitar el uso del paralelismo GPU en sus tarjetas gráficas. Las tarjetas gráficas Nvidia Gforce, orientadas a uso domestico, pueden utilizar CUDA para aprovechar su procesamiento paralelo. La plataforma de cálculo paralelo CUDA proporciona un conjunto de abstracciones que permiten implementar el paralelismo en el procesamiento de tareas y procesos con diferentes niveles de granularidad. El programador puede expresar este paralelismo mediante diferentes lenguajes de alto nivel como C++ o Fortran.

Para hacernos una idea del coste, a junio de 2012, una Gforce GT 640 que ofrece 384 núcleos a una velocidad de 625Mhz con una memoria dedicada de 2GB DDR3 puede estar sobre los sobre

los 120€ (el precio y la cantidad de memoria disponible en la tarjeta depende de la empresa ensambladora: ASUS, Gigabyte, Point View, etc.).

Actualmente podemos encontrar tarjetas para paralelismo extremo en equipos de escritorio, utilizadas principalmente para juegos, como la Gforce GTX Titan X que ofrece 3072 núcleos, con una memoria dedicada de 12GB DDR5 y cuesta sobre los 1.000€.

Además Nvidia ha desarrollado la serie de tarjetas denominada Tesla (que sustituyen a las Fermi) orientadas directamente al “GPU-Computing” o cálculo científico GP-GPU, se venden directamente premontadas en equipos de trabajo o en equipos enrackables para formar nodos de supercomputación. En este caso para equipos de trabajo o servidores con procesadores Intel Xeon estaríamos hablando a partir de 3.000€, de una GPU Tesla C2075 con 448 núcleos, 6GB de memoria dedicada DDR5, 1,03 Tflops y un ancho de banda mucho mayor que sus hermanas pequeñas Gforce. Actualmente se encuentran disponibles los modelos Tesla ‘K20’ (2.496 núcleos, 5GB DDR5 y 1.17 Tflops) y ‘K40’ (2.880 núcleos, 12 GB DDR5 y 1,43 Tflops) para entornos de supercomputación, con un valor entre 5.000€ y 10.000€.

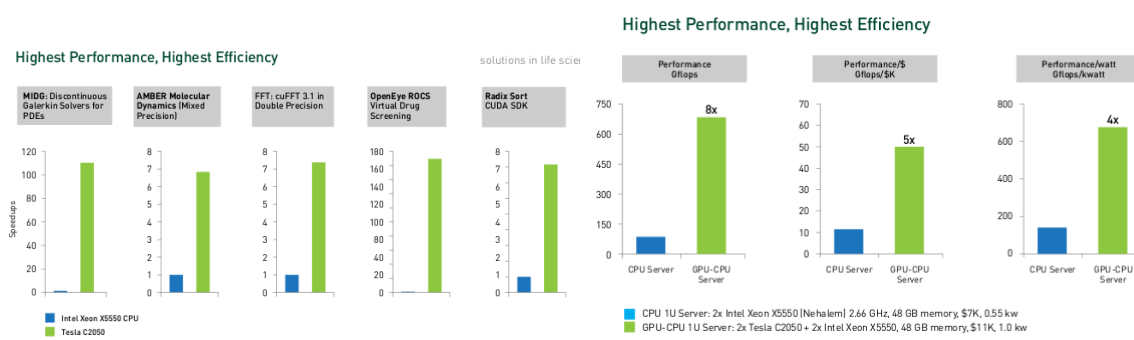


Figura G.2: Comparativa del “speed-up” y “performance” entre CPU y GPU, para diferentes herramientas (<http://www.nvidia.es/object/gpu-computing-es.html>).

Aplicación de las GPUs al SPH En el caso que nos ocupa, el objetivo es implementar o utilizar algoritmos en GPUs para la resolución de sistemas de ecuaciones con matrices dispersas. La estructura interna de la GPU difiere de la CPU tradicional y por tanto es necesario un nuevo planteamiento en el tratamiento de los problemas. A diferencia de las CPUs, que tienen gran versatilidad y un propósito general, para usar las GPUs es necesario un estudio previo de problema para identificar los puntos adecuados donde puede ser útil la utilización del paralelismo GPU. Las GPUs por si solas no permiten ejecutar en el sistema un programa, ni sirven para todos los tipos de cálculos. Para su ejecución en el sistema un programa necesita del uso de la CPU, es tarea del programador seleccionar que parte del mismo puede ser implementada para su ejecución en paralelo en las GPUs. Por tanto los códigos implementados serán un compromiso

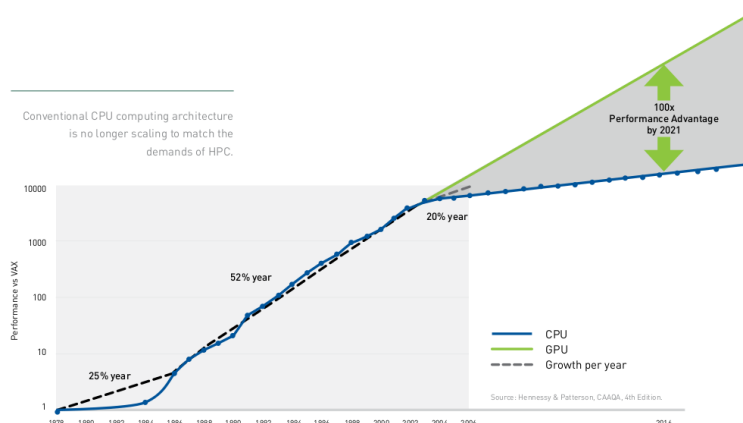


Figura G.3: Comparativa de la proyección entre CPU y GPU, para el futuro (<http://www.nvidia.es/object/gpu-computing-es.html>).

entre el uso de CPUs y GPUs, en función de las necesidades que se observen y siempre tratando de minimizar el empleo de la CPU para evitar las transferencias de datos costosas que se producen usando el bus entre la memoria principal y la memoria dedicada de la tarjeta. Podemos encontrar una buena introducción a la aplicabilidad de GPUs en SPH en *Rey, 2010* [76].

GPUs en el top500 En el año 2.007 aparece el primer supercomputador basado en tecnología GPU que gracias a sus 77,48 TFlops se sitúa en el puesto 29 de la lista top500 que rankea los mejores supercomputadores en función de su capacidad de cálculo. Esta máquina es *TSUBAME* del “Tokyo Institute of Technology”, que cuenta con 170 Nvidia Tesla S1070 1U y una capacidad de cálculo usando las tarjetas gráficas de 170 TFlops. Desde entonces una multitud de máquinas del top500 usan tecnología GPU. Por ejemplo, ya en noviembre de 2011 aparecen en la lista 35 máquinas aceleradas por GPU Tesla, siendo 14 de ellas de nueva inclusión y 3 de ellas situándose en el top5. El crecimiento exponencial en el número de supercomputadores GPU en la lista top500 es una de las mas rápidas adaptaciones de un nuevo tipo de procesadores de la historia de la informática de alto rendimiento (HPC).

Intel Xeon Phi

Como alternativa a las tarjetas gráficas en 2012 Intel lanzó al mercado una nueva gama de productos, las tarjetas PCI-Express de co-procesadores Intel Xeon Phi. Su principal ventaja es que tiene un gran ancho de banda y utiliza memoria caché (a diferencia de las GPUs) y por tanto puede hacer paralelización de datos (SIMD). Las tarjetas de co-procesadores actúan como nodos internos de la máquina (pudiendo hacer ssh a la tarjeta). Es posible montar múltiples tarjetas

de coprocesadores en un mismo sistema. Pueden funcionar en dos modos: 1) Nativo, ejecutando directamente en la tarjeta o 2) Mixto, la CPU ejecuta el programa y distribuye el trabajo entre las diferentes tarjetas Intel Xeon Phi, mediante comunicaciones por el bus del sistema.



Figura G.4: Intel Xeon Phi: Knight Corner - Múltiples tarjetas.

La primera versión en salir al mercado fue *Knights Corner*, con 60 cores, 240 threads, 6-8GB DDR5 y con una potencia de cálculo de 1 Tflops (similar a la NVidia Tesla K20) y un coste, dependiendo de las versiones, entre los 2.500 € y los 5.000 €. Es necesario compilar específicamente para esta arquitectura y añadir directivas de comunicación en caso de uso mixto. La comunicación entre los diferentes cores de la tarjeta se realiza mediante un bus en anillo, reduciendo así los tiempos de acceso a datos.

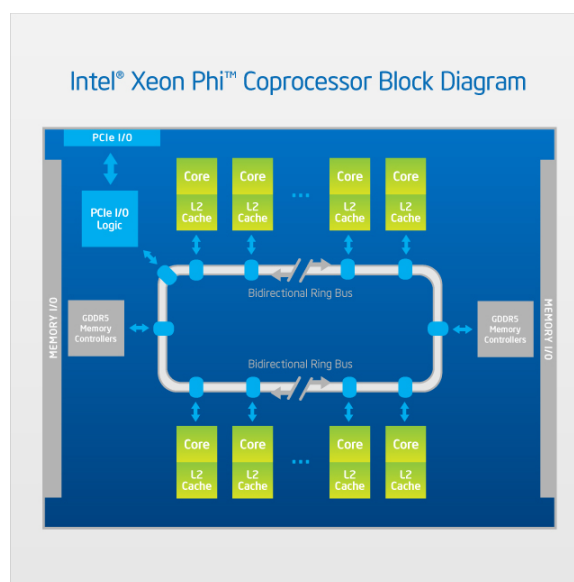


Figura G.5: Intel Xeon Phi: Knight Corner - Diagrama.

La nueva versión denominada *Knights Landing*, disponible a principios de 2016, es compa-

tible con la arquitectura Intel Xeon (porque el vector de registros es similar AVX-512). No es necesario recompilar para la arquitectura, aunque si es recomendable para aprovechar las nuevas características. La comercialización se realizará en tarjetas de co-procesadores, para sistemas con Intel Xeon y como nodos independientes, que parece ser la apuesta de Intel. A diferencia de la versión anterior, utilizará un bus matricial de comunicación entre cores que resulta mucho mas eficiente.

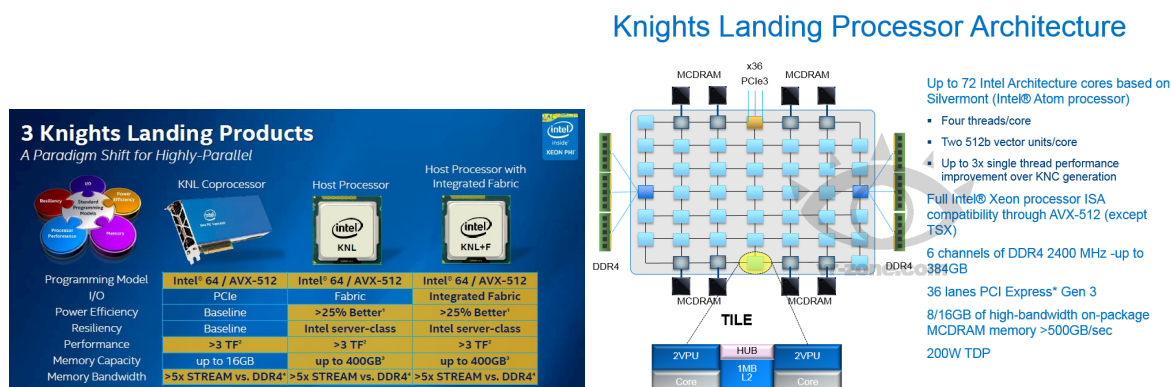


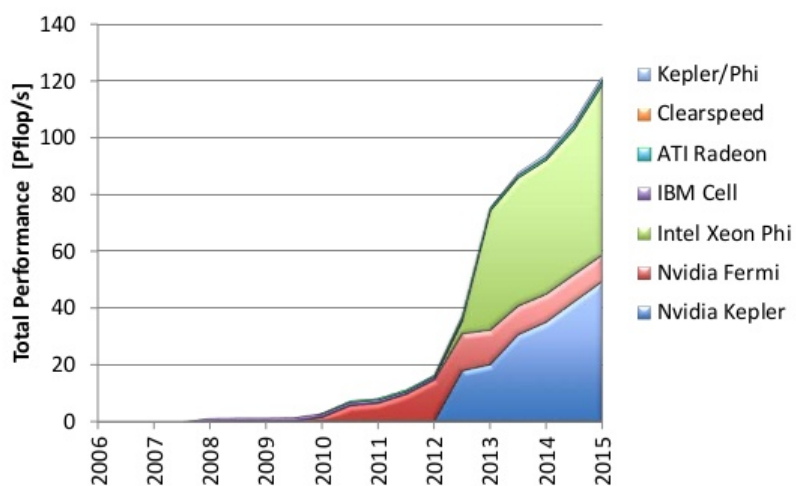
Figura G.6: Intel Xeon Phi: Knight Landing - Diagrama.

Clusters: Top500 44th(2014) y 45th(2015)

#	Site	Manufacturer	Computer	Country	Cores	P _{max} (preprod)	Power (MW)
1	National University of Defense Technology	NUDT	Tianhe-2 NUDT TH-IVB-FEP, Xeon 12C 2.2GHz, Intel Xeon Phi	China	3,120,000	33.9	17.8
2	Oak Ridge National Laboratory	Cray	Titan Cray XK7, Opteron 16C 2.2GHz, Gemini, NVIDIA K20x	USA	560,640	17.6	8.21
3	Lawrence Livermore National Laboratory	IBM	Sequoia BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	1,572,864	17.2	7.89
4	RIKEN Advanced Institute for Computational Science	Fujitsu	K Computer SPARC64 VIIIfx 2.0GHz, Tofu Interconnect	Japan	795,024	10.5	12.7
5	Argonne National Laboratory	IBM	Mira BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	786,432	8.59	3.95
6	Swiss National Supercomputing Centre (CSCS)	Cray	Piz Daint Cray XC30, Xeon E5 8C 2.6GHz, Aries, NVIDIA K20x	Switzerland	115,984	6.27	2.33
7	King Abdullah University of Science and Technology	Cray	Shaheen II Cray XC40, Xeon E5 16C 2.3GHz, Aries	Saudi Arabia	196,608	5.54	2.83
8	Texas Advanced Computing Center/UT	Dell	Stampede PowerEdge C8220, Xeon E5 8C 2.7GHz, Intel Xeon Phi	USA	462,462	5.17	4.51
9	Forschungszentrum Juelich (FZJ)	IBM	JuQUEEN BlueGene/Q, Power BQC 16C 1.6GHz, Custom	Germany	458,752	5.01	2.30
10	Lawrence Livermore National Laboratory	IBM	Vulcan BlueGene/Q, Power BQC 16C 1.6GHz, Custom	USA	393,216	4.29	1.97

Figura G.7: Lista *top500* 45th(2015): List (<http://www.slideshare.net/top500/top500-201506>).

PERFORMANCE OF ACCELERATORS



PERFORMANCE SHARE OF ACCELERATORS

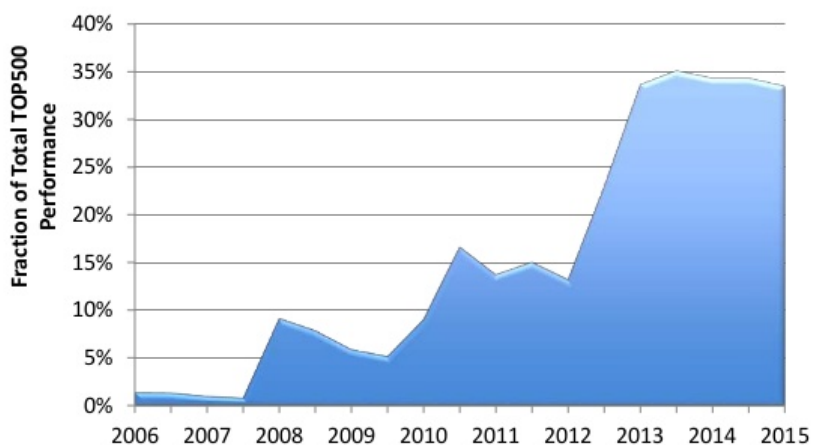


Figura G.8: Lista *top500* 45th(2015): Distribución de aceleradoras (<http://www.slideshare.net/top500/top500-201506>).

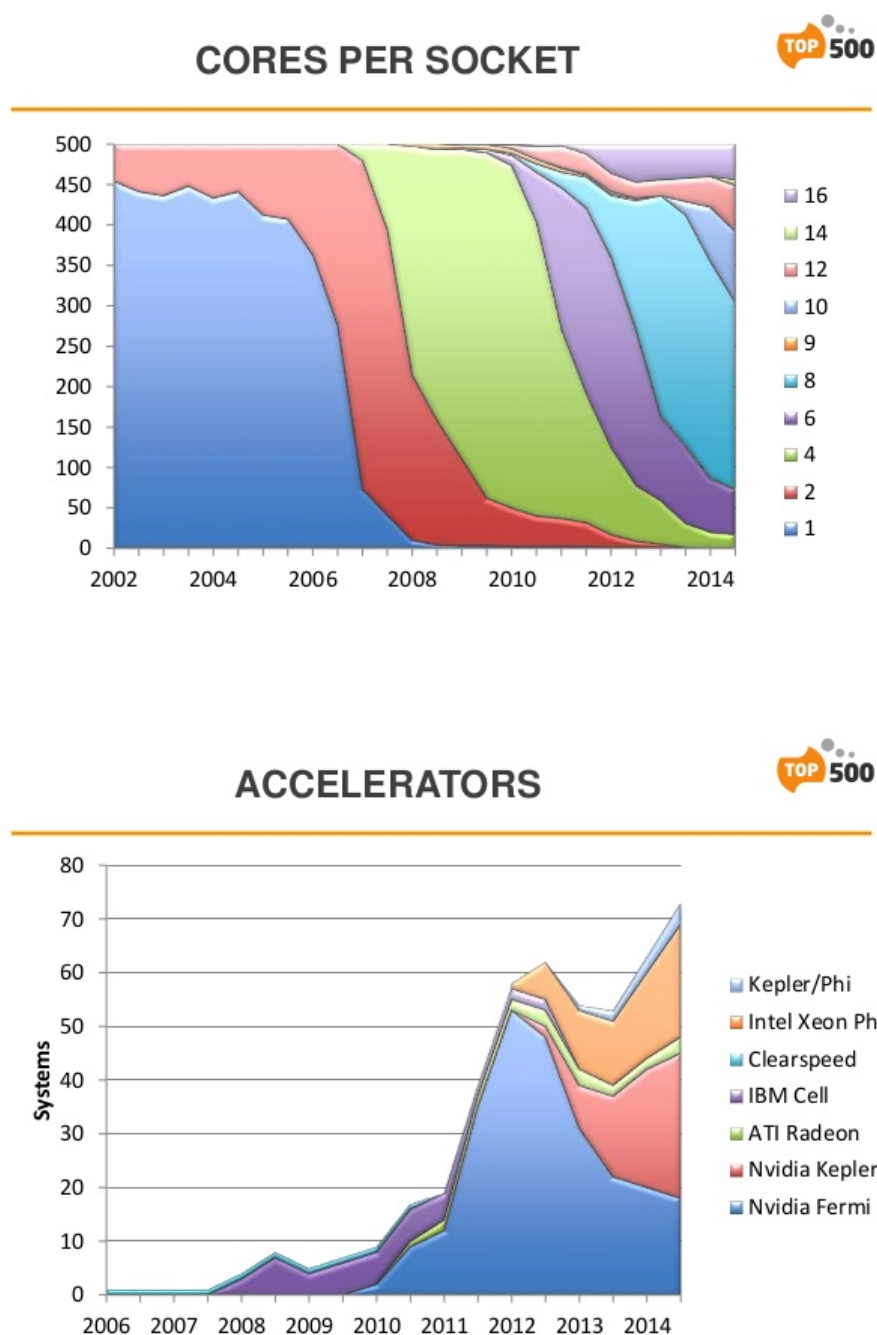


Figura G.9: Lista *top500* 44th(2014): Núcleos por socket y distribución de aceleradoras (<http://www.slideshare.net/top500/top500-list-november-2014>).

Apéndice H

Publicaciones

Han sido varios los textos que se han publicado a partir del trabajo de estos años. Algunos están directamente relacionados con él y otros tienen su centro en varios aspectos del SPH y las simulaciones numéricas. Los detallo a continuación según las áreas objetivo:

Código SPH implícito ISFAA

- Escartín, J.A.; García-Senz, D.; “ISFAA, prospects for an implicit Smoothed Particle Hydrodynamics”, Highlights of Spanish astrophysics VII, Proceedings; p.943, 2013.
- Escartín, J.A., García-Senz, D., Cabezón, R.M. “ISFAA, prospects for an implicit SPH”, 6th international Spheric workshop; p.194-200, 2011, Schriftenreihe Schiffbau ISBN 978-3-89220-658-3.
- Escartín, J.A.; García-Senz, D.; Bravo, E. “Computer simulations using implicit Lagrangian hydrodynamics in 3D”, Highlights of Spanish astrophysics VI, Proceedings; p.703-708, 2011.
- Escartín, J.A.; “Hidrodinámica lagrangiana en 2D: Aproximación implícita”, Master Thesis, 2009: <http://upcommons.upc.edu/pfc/handle/2099.1/6626>.

Estudio de reacciones nucleares en Supernovas Ia

- Bravo, E.; Piersanti, L.; Domínguez, I.; Straniero, O.; Isern, J.; Escartín, J.A.; “Type Ia supernovae and the $^{12}\text{C}+^{12}\text{C}$ reaction rate”, Astronomy & Astrophysics 535 p.A114, 2011.

Ampliación de los Kernels Sinc - Índice adaptativo del kernel

- García-Senz D., Cabezón R.M., Escartín J.A., Ebinger K.; “Equalizing resolution in smoothing particle hydrodynamics calculations using self-adaptive sinc kernels”, *Astronomy & Astrophysics* 570 p.A14, 2014.

Modificación tensorial del SPH estándar (IAD)

- García-Senz, D.; Cabezón, R.M.; Escartín, J.A.; “Improving the smoothed particle hydrodynamics with an integral approach to calculating gradients”, *Astronomy & Astrophysics* 538 p.A9, 2012.
- Cabezón, R.M.; García-Senz, D.; Escartín, J.A.; “Testing the concept of integral approach to derivatives within the smoothed particle hydrodynamics technique in astrophysical scenarios”, *Astronomy & Astrophysics* 545 p.A112, 2012.
- Escartín, J.A., García-Senz, D., Cabezón, R.M. “Smoothed particle hydrodynamics: Checking a tensor approach to calculating gradients”, *Highlights of Spanish astrophysics VII, Proceedings*; p.874-879, 2013.

Software de visualización de gráficos para simulaciones numéricas

- Escartin, J.A., Garcia-Senz, D. “A displayer of stellar hydrodynamics processes”, *Highlights of Spanish astrophysics V*; p. 497-498, 2010, Springer ISBN 978-3-642-11249-2.

Bibliografía

- [1] ALMGREN, A. S., BELL, J. B., NONAKA, A., AND ZINGALE, M. Low mach number modeling of type ia supernovae. iii. reactions. *ApJ* 684 (2008), 449–470.
- [2] ALMGREN, A. S., BELL, J. B., RENDLEMAN, C. A., AND ZINGALE, M. Low mach number modeling of type ia supernovae. i. hydrodynamics. *ApJ* 637 (2006), 922–936.
- [3] ALMGREN, A. S., BELL, J. B., RENDLEMAN, C. A., AND ZINGALE, M. Low mach number modeling of type ia supernovae. ii. energy evolution. *ApJ* 649 (2006), 927–938.
- [4] ATZENI, S. Letter: Ablative stabilization of short wavelength rayleigh-taylor instability. *Nuclear Fusion* 36 (1996), 69–74.
- [5] ATZENI, S., AND MEYER-TER-VEHN, J. *The Physics of Inertial Fusion: Beam Plasma Interaction, Hydrodynamics, Hot Dense Matter*. Oxford Science Publications, 2004.
- [6] AZNAR-SIGUÁN, G., GARCÍA-BERRO, E., LORÉN-AGUILAR, P., JOSÉ, J., AND ISERN, J. Detonations in white dwarf dynamical interactions. *MNRAS* 434 (2013), 2539–2555.
- [7] BALSARA, D. S. Von neumann stability analysis of smoothed particle hydrodynamics - suggestions for optimal algorithms. *J. Comp. Phys.* 121 (1995), 357–372.
- [8] BARNES, J., AND HUT, P. A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature* 324 (1986), 446–449.
- [9] BENZ, W. Smooth particle hydrodynamics: Theory and application to the origin of the moon. *NASA STI/Recon Technical Report 87* (1986), 14235.
- [10] BERCZIK, P. Modeling the star formation in galaxies using the chemo-dynamicsph code. *Astrophysics and Space Science* 271 (2000), 103–126.
- [11] BERGER, M. J., AND OLIGER, J. E. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comp. Phys.* 53 (1984), 484.

- [12] BLINNIKOV, S. I., PANOV, I. V., RUDZSKY, M. A., AND SUMIYOSHI, K. The equation of state and composition of hot, dense matter in core-collapse supernovae. *A&A* 535 (2011), A37.
- [13] BRAVO, E., AND GARCÍA-SENZ, D. A three-dimensional picture of the delayed-detonation model of type ia supernovae. *A&A* 478 (2008), 843–853.
- [14] CABEZÓN, R. M. *Hydrodynamic simulations of DNS systems: Gravitational emission and equation of state*. PhD thesis, Universidad Politécnica de Cataluña, 2010.
- [15] CABEZON, R. M., AND GARCÍA-SENZ, D. Gravitational waves as tracers of nuclear equation of state. In *Supernovae: lights in the darkness* (2007).
- [16] CABEZÓN, R. M., GARCÍA-SENZ, D., AND ESCARTÍN, J. A. Testing the concept of integral approach to derivatives within the smoothed particle hydrodynamics technique in astrophysical scenarios. *A&A* 545 (2012), A112.
- [17] CABEZÓN, R. M., GARCÍA-SENZ, D., AND RELAÑO, A. A one-parameter family of interpolating kernels for smoothed particle hydrodynamics studies. *J. Comp. Phys.* 227 (2008), 8523–8540.
- [18] CAMENZIND, M. Numerical magnetohydrodynamics in astrophysics. In *Cosmic Magnetic Fields* (2005), vol. 664 of *Lecture Notes in Physics*, Berlin Springer Verlag, p. 255.
- [19] CHA, S. H., AND WHITWORTH, A. P. Implementations and tests of godunov-type particle hydrodynamics. *MNRAS* 340 (2003), 73–90.
- [20] CHANDRASEKHAR, S. *Hydrodynamic and Hydromagnetic Stability*. Dover Publications, 1981.
- [21] CLAYTON, D. D. *Principles of stellar evolution and nucleosynthesis*. McGraw-Hill, 1968.
- [22] COURANT, R., FRIEDRICHS, K., AND LEWY, H. On the partial differential equations of mathematical physics. *IBM J. Res. Dev.*, 11 (1967), 215–234.
- [23] DAHLQUIST, G., AND BJÖRCK, A. *Numerical Methods*. Dover Publications, 2003.
- [24] DE LA FUENTE, J. *Técnicas de Cálculo para Sistemas de Ecuaciones, Programación Lineal y Programación Entera*. Reverté, 1997.
- [25] EVRARD, A. E. Beyond n-body - 3d cosmological gas dynamics. *MNRAS* 235 (1988), 911–934.
- [26] GARCÍA-SENZ, D., CABEZÓN, R. M., AND ESCARTÍN, J. A. Improving smoothed particle hydrodynamics with an integral approach to calculating gradients. *A&A* 538 (2012), A9.

- [27] GARCÍA-SENZ, D., RELAÑO, A., CABEZÓN, R. M., AND BRAVO, E. Axisymmetric smoothed particle hydrodynamics with self-gravity. *MNRAS* 392 (2009), 346–360.
- [28] GLASNER, S. A., LIVNE, E., AND TRURAN, J. W. Novae: The evolution from onset of convection to the runaway. *ApJ* 665 (2007), 1321–1333.
- [29] GÓMEZ-GESTEIRA, M., DALRYMPLE, R. A., CRESPO, A. J. C., AND CERQUEIRO, D. Uso de la técnica sph para el estudio de la interacción entre olas y estructuras. *Ingeniería del agua* 11 (2004), 2.
- [30] GOULD, N. I., SCOTT, J. A., AND HU, Y. A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Trans. Math. Softw.* 33 (2007).
- [31] GUERRERO, J., GARCÍA-BERRO, E., AND ISERN, J. Smoothed particle hydrodynamics simulations of merging white dwarfs. *A&A* 413 (2004), 257–272.
- [32] GUPTA, A. *WSMP: Watson sparse matrix package (Part-III: Iterative solution of sparse systems)*.
- [33] GUPTA, A. *WSMP: Watson sparse matrix package (Part II: direct solution of general sparse systems)*, 2000. Modified 2011.
- [34] GUPTA, A. Recent advances in direct methods for solving unsymmetric sparse systems of linear equations. *ACM Trans. Math. Softw.* 28 (2002), 301–324.
- [35] GUPTA, A., KORIC, S., AND GEORGE, T. Sparse matrix factorization on massively parallel computers. In *SC '09 Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis* (2009), pp. 1:1–12.
- [36] HENYEV, L., FORBES, J. E., AND GOULD, N. L. A new method of automatic computation of stellar evolution. *ApJ* 139 (1964), 306.
- [37] HENYEV, L. G., WILETS, L., BÖHM, K. H., LELEVIER, R., AND LEVEE, R. D. A method for automatic computation of stellar evolution. *ApJ* 129 (1959), 628.
- [38] HERNQUIST, L., AND KATZ, N. Treesph - a unification of sph with the hierarchical tree method. *ApJS* 70 (1989), 419–446.
- [39] HILLEBRANDT, W., AND NIEMEYER, J. C. Type ia supernova explosion models. *ARA&A* 38 (2000), 191–230.
- [40] HÖFLICH, P., AND STEIN, J. On the thermonuclear runaway in type ia supernovae: How to run away? *ApJ* 568 (2002), 779–790.

- [41] HOPKINS, P. F. A new class of accurate, mesh-free hydrodynamic simulation methods. *MNRAS* 450 (2015), 53–110.
- [42] INUTSUKA, S. I. Reformulation of smoothed particle hydrodynamics with riemann solver. *J. Comp. Phys.* 179 (2002), 238–267.
- [43] JOHNSON, G. R., STRYK, R. A., AND R., B. S. Sph for high velocity impact computations. *Computer Methods in Applied Mechanics and Engineering* 139 (1996), 347–373.
- [44] JUBELGAS, M., SPRINGEL, V., AND DOLAG, K. Thermal conduction in cosmological sph simulations. *MNRAS* 351 (2004), 423–435.
- [45] KERCEK, A., HILLEBRANDT, W., AND TRURAN, J. W. Three-dimensional simulations of classical novae. *A&A* 345 (1999), 831–840.
- [46] KIFONIDIS, K., AND MÜLLER, E. On multigrid solution of the implicit equations of hydrodynamics. experiments for the compressible euler equations in general coordinates. *A&A* 544 (2012), A47.
- [47] KNAPP, C. E. *An Implicit Smooth Particle Hydrodynamic Code*. PhD thesis, University of New Mexico (Los Alamos National Laboratory), 2000.
- [48] LIEBENDÖRFER, M., ROSSWOG, S., AND THIELEMANN, F.-K. An adaptive grid, implicit code for spherically symmetric, general relativistic hydrodynamics in comoving coordinates. *ApJS* 141 (2002), 229–246.
- [49] LIN, D. J., BAYLISS, A., AND TAAM, R. E. Low mach number modeling of type i x-ray burst deflagrations. *ApJ* 653 (2006), 545–557.
- [50] LINDL, J. Development of the indirect-drive approach to inertial confinement fusion and the target physics basis for ignition and gain. *Phys. of Plasmas* 2 (1995), 3933–4024.
- [51] LOBATCHEV, V., AND BETTI, R. Ablative stabilization of the deceleration phase rayleigh-taylor instability. *Phys. Rev. Lett.* 85 (2000), 4522–4525.
- [52] LORÉN-AGUILAR, P. *Gravitational wave radiation from single and binary white dwarfs*. PhD thesis, Universidad Politécnica de Cataluña, 2010.
- [53] LORÉN-AGUILAR, P., ISERN, J., AND GARCÍA-BERRO, E. High-resolution smoothed particle hydrodynamics simulations of the merger of binary white dwarfs. *A&A* 500 (2009), 1193–1205.
- [54] LUCY, L. B. Numerical approach to the testing of the fission hypothesis. *AJ* 82 (1977), 1013–1024.

- [55] MACNEICE, P., OLSON, K. M., MOBARRY, C., DE FAINCHEIN, R., AND PACKER, C. Paramesh: A parallel adaptative mesh refinement community toolkit. *Comp. Phys. Com.* 126 (2000), 330–354.
- [56] MENÉNDEZ, A. N. Errores en la solución numérica de ecuaciones diferenciales. Tech. rep., Instituto nacional de ciencia y técnicas hídricas, Reedición 2010.
- [57] MONAGHAN, J. J. On the problem of penetration in particle methods. *J. Comp. Phys.* 82 (1989), 1–15.
- [58] MONAGHAN, J. J. Smoothed particle hydrodynamics. *ARA&A* 30 (1992), 543–574.
- [59] MONAGHAN, J. J. Simulating free surface flows with sph. *J. Comp. Phys.* 110 (1994), 399–406.
- [60] MONAGHAN, J. J. Sph and riemann solvers. *J. Comp. Phys.* 136 (1997), 298–307.
- [61] MONAGHAN, J. J. Smoothed particle hydrodynamics. *Rep. Prog. Phys.* 68 (2005), 1703–1759.
- [62] MONAGHAN, J. J., AND GINGOLD, R. A. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *MNRAS* 181 (1977), 375–389.
- [63] MONAGHAN, J. J., AND LATTANZIO, J. C. A refined particle method por astrophysical problems. *A&A* 149 (1985), 135–143.
- [64] MORTON, G. M. A computer oriented geodetic data base; and a new technique in file sequencing. Tech. rep., IBM Ltd., 1966.
- [65] MORTON, K. W., AND MAYERS, D. F. *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press, 2005.
- [66] NANDKUMAR, R., AND PETHICK, C. J. Transport coefficients of dense matter in the liquid metal regime. *MNRAS* 209 (1984), 511.
- [67] NOH, W. F. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *J. Comp. Phys.* 72 (1987), 78–120.
- [68] OGANDO, F., AND VELARDE, P. Development of a radiation transport fluid dynamic code under amr scheme. *J. Quant. Spec. Radiat. Transf.* 71 (2001), 541–550.
- [69] OWEN, J. M., VILLUMSEN, J. V., SHAPIRO, P. R., AND MARTEL, H. Adaptive smoothed particle hydrodynamics: Methodology. ii. *ApJS* 116 (1998), 155–209.
- [70] PEREGO, A., ROSSWOG, S., CABEZÓN, R. M., KOROBKIN, O., KÄPPELI, R., ARCONES, A., AND LIEBENDÖRFER, M. Neutrino-driven winds from neutron star merger remnants. *MNRAS* 443 (2014), 3134–3156.

- [71] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [72] PRICE, D. J., AND BATE, M. R. The impact of magnetic fields on single and binary star formation. *MNRAS* 377 (2007), 77–90.
- [73] RELAÑO, A. *AxiSPH: Devising and validating an axisymmetric smoothed particle hydrodynamics code*. PhD thesis, Universidad Politécnica de Cataluña, 2012.
- [74] RELAÑO, A., GARCÍA-SENZ, D., AND BRAVO, E. Numerical hydrodynamics of inertial confinement fusion: First steps toward a gridless particle approach. In *Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications* (1999), pp. 228–238.
- [75] RELAÑO, A., GARCÍA-SENZ, D., AND BRAVO, E. The stagnation phase of an icf capsule simulated with an axisymmetrical smoothed-particle-hydrodynamics code. In *In Proceedings of the XXIX ECLIM 29th European Conference on Laser Interaction with Matter* (2006), pp. 91–96.
- [76] REY, A. A. Implementación del método de partículas sph para la simulación de fluidos sobre tarjetas gráficas. Master’s thesis, Escuela técnica superior de ingeniería (ICAI), 2010.
- [77] RICHTMYER, R. D., AND MORTON, K. W. *Difference methods for initial value problems*. John Wiley, 1967.
- [78] ROSSWOG, S. Astrophysical smooth particle hydrodynamics. *New Astronomy Rev.* 53 (2009), 78–104.
- [79] ROSSWOG, S. Conservative, special relativistic smoothed particle hydrodynamics. *J. Comp. Phys.* 229 (2010), 8591–8612.
- [80] ROSSWOG, S., AND PRICE, D. Magma: a three-dimensional, lagrangian magnetohydrodynamics code for merger applications. *MNRAS* 379 (2007), 915–931.
- [81] SAKAGAMI, H., AND NISHIHARA, K. Rayleigh-taylor instability on the pusher-fuel contact surface of stagnating targets. *Physics of Fluids B* 2 (1990), 2715–2730.
- [82] SAKAIYA, T., AZECHI, H., MATSUOKA, M., IZUMI, N., NAKAI, M., SHIGEMORI, K., SHIRAGA, H., SUNAHARA, A., TAKABE, H., AND YAMANAKA, T. Ablative rayleigh-taylor instability at short wavelengths observed with moiré interferometry. *Phys. Rev. Lett.* 88, 14 (2002), 145.
- [83] SCHENK, O., BOLLHÖFER, M., AND RÖMER, R. A. On large-scale diagonalization techniques for the anderson model of localization. *SIAM Rev.* 50 (2008), 91–112.

- [84] SCHENK, O., AND GÄRTNER, K. Solving unsymmetric sparse systems of linear equations with pardiso. *Future Gener. Comput. Syst.* 20 (2004), 475–487.
- [85] SCHENK, O., AND GÄRTNER, K. On fast factorization pivoting methods for sparse symmetric indefinite systems. *Electronic Transactions on Numerical Analysis* 23 (2006), 158–179.
- [86] SCHENK, O., WÄCHTER, A., AND HAGEMANN, M. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Comput. Optim. Appl.* 36 (2007), 321–341.
- [87] SCHINDER, P. J., AND BLUDMAN, S. A. Radiative transport in spherical static spacetime - general relativistic tangent-ray method for the variable eddington factors. *ApJ* 346 (1989), 350–365.
- [88] SCHOENBERG, I. J., AND SHARMA, A. Cardinal interpolation and spline functions v. the b-splines for cardinal hermite interpolation. *Linear Algebra Appl.* 7 (2073), 1–42.
- [89] SEDOV, L. I. *Similarity and Dimensional Methods in Mechanics*. Academic Press, 1959.
- [90] SERICHOL, N. *Interacció del material expulsat en les explosions de supernova amb diferents tipus d'estels acompanyants*. PhD thesis, Universidad Politécnica de Cataluña, 2005.
- [91] SERNA, A., ALIM, J.-M., AND CHIEZE, J.-P. Adaptive smooth particle hydrodynamics and particle-particle coupled codes: Energy and entropy conservation. *ApJ* 461 (1996), 884.
- [92] SMALYUK, V. A., DELETTREZ, J. A., GONCHAROV, V. N., MARSHALL, F. J., MEYER-HOFER, D. D., REGAN, S. P., SANGSTER, T. C., TOWN, R. P. J., AND YAAKOBI, B. Rayleigh-taylor instability in the deceleration phase of spherical implosion experiments. *Physics of Plasmas* 9 (2002), 2738–2744.
- [93] SPRINGEL, V. The cosmological simulation code gadget-2. *MNRAS* 364 (2005), 1105–1134.
- [94] SPRINGEL, V. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *MNRAS* 401 (2010), 791–851.
- [95] SPRINGEL, V. Smoothed particle hydrodynamics in astrophysics. *ARA&A* 48 (2010), 391–430.
- [96] SPRINGEL, V., AND HERNQUIST, L. Cosmological smoothed particle hydrodynamics simulations: the entropy equation. *MNRAS* 333 (2002), 649–664.
- [97] STEINMETZ, M. Grapesph: cosmological smoothed particle hydrodynamics simulations with the special-purpose hardware grape. *MNRAS* 278 (1996), 1005–1017.

-
- [98] SWESTY, F. D. Implicit general relativistic hydrodynamic methods for modeling the late-time explosion mechanism in core collapse supernovae. *ApJ* 445 (1995), 811–827.
- [99] TOURNIER, J. C., DONDE, V., AND LI, Z. Comparison of direct and iterative sparse linear solvers for power system applications on high performance computing platforms. In *17th Power Systems Computation Conference. Stockholm, Sweden* (2011).
- [100] VIALLET, M., BARAFFE, I., AND WALDER, R. Towards a new generation of multi-dimensional stellar evolution models: development of an implicit hydrodynamic code. *A&A* 531 (2011), A86.
- [101] WADSLEY, J. W., STADEL, J., AND QUINN, T. Gasoline: a flexible, parallel implementation of treesph. *New Astronomy* 9 (2004), 137–158.
- [102] WIERSMA, R. P. C., SCHAYE, J., THEUNS, T., DALLA VECCHIA, C., AND TORNATORE, L. Chemical enrichment in cosmological, smoothed particle hydrodynamics simulations. *MNRAS* 399 (2009), 574–600.
- [103] YAMADA, S. An implicit lagrangian code for spherically symmetric general relativistic hydrodynamics with an approximate riemann solver. *ApJ* 475 (1997), 720.
- [104] ZEL'DOVICH, B., AND RAIZER, P. *Physics of shock waves and high-temperature hydrodynamic phenomena*, vol. 2. Academic Press, 1967.
- [105] ZHANG, W., AND MACFADYEN, A. I. Ram: A relativistic adaptive mesh refinement hydrodynamics code. *ApJS* 164 (2006), 255–279.
- [106] ZINGALE, M., ALMGREN, A. S., BELL, J. B., NONAKA, A., AND WOOSLEY, S. E. Low mach number modeling of type ia supernovae. iv. white dwarf convection. *ApJ* 704 (2009), 196–210.